

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial
15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

MAESTRÍA EN DISEÑO ELECTRÓNICO



REPORTE DE FORMACIÓN COMPLEMENTARIA EN ÁREA DE CONCENTRACIÓN EN SISTEMAS EMBEBIDOS Y TELECOMUNICACIONES

Trabajo recepcional que para obtener el grado de
MAESTRA EN DISEÑO ELECTRÓNICO

Presenta: Liliana Marcela Álvarez de la Cruz

Asesor: Dr. Jose Luis Chavez Hurtado

San Pedro Tlaquepaque, Jalisco. Noviembre de 2017.

MAESTRO EN INGENIERÍA (2017)

Maestría en Diseño Electrónico

ITESO

Tlaquepaque, Jal., México

ÁREA DE CONCENTRACIÓN: “Sistemas Embebidos y Telecomunicaciones”

AUTOR:

Liliana Marcela Alvarez de la Cruz

Ingeniera en Comunicaciones y Electrónica (Universidad de
Guadalajara, México)

REVISORES:

José Luis Chávez Hurtado

NÚMERO DE PÁGINAS:

v, 85

Contenido

Instituto Tecnológico y de Estudios Superiores de Occidente	i
1. Resumen de los proyectos realizados	3
1.1. SISTEMAS EMBEBIDOS AVANZADOS.....	3
1.1.1 Introducción	3
1.1.2 Antecedentes	4
1.1.3 Solución Desarrollada	5
1.1.4 Análisis de Resultados	6
1.2. TALLER DE DISEÑO DE PCBs	7
1.2.1 Introducción	7
1.2.2 Antecedentes	8
1.2.3 Solución Desarrollada	8
1.2.4 Análisis de Resultados	10
1.3. PROCESAMIENTO DIGITAL DE SEÑALES	11
1.3.1 Introducción	11
1.3.2 Antecedentes	11
1.3.3 Solución Desarrollada	12
1.3.4 Análisis de Resultados	14
1.4. CONCLUSIONES GENERALES	14
Apéndices	17
A. SISTEMAS EMBEBIDOS AVANZADOS	18
B. TALLER DE DISEÑO DE TARJETAS DE CIRCUITO IMPRESO.....	38
C. PROCESAMIENTO DIGITAL DE SEÑALES	82
1. Introduction.....	83
2. Algorithms	83
2.1. COLOR REDUCCION	83
2.2. EDGES DETECTION.....	84
2.3. EDGES DISTORTION	84
3. RESULTS	85

Introducción

La maestría en diseño electrónico fue el complemento de mis estudios en Ingeniería en electrónica y comunicaciones para seguir con mi formación académica, Para titularme de la maestría escogí la modalidad de titulación por área de concentración.

Para este trabajo escogí el área de concentración de Sistemas Embebidos y Telecomunicaciones debido a de las actividades profesionales que desarrolla en mi empleo en ese tiempo y aprovechando las herramientas que tenía a la mano para poder desarrollar mis investigaciones y prácticas.

Las materias seleccionadas y en base a lo que se realizaron los proyectos del área de concentración son:

- Sistemas Embebidos Avanzados.
- Taller de Diseño de PCBs.
- Procesamiento Digital de Señales

Como ingeniera de Validación (posición en Intel), mis responsabilidades eran hacer pruebas tanto para validar el correcto funcionamiento de diferentes módulos de microprocesadores como las conexiones en las tarjetas que simulan el comportamiento de un producto final.

La materia de sistemas embebidos me ayudo con la comprensión de la comunicación entre componentes.

La materia de Diseño de PCBs me ayudo a incrementar mi creatividad para hacer mejores test plan para validar las tarjetas.

La materia Digital de Señales la tomo como recreación ya que esta interesante el tema y podría servirme para la validación de los módulos encargados de gráficos.

1. Resumen de los proyectos realizados

Dentro de las materias que cursé en el área de especialización de sistemas embebidos y telecomunicaciones, realicé proyectos en cada una de las materias los cuales fueron la culminación de la mismas y medirían el aprovechamiento de los recursos y conocimientos que obtuve durante la duración del curso.

En la materia de sistemas embebidos avanzados implementamos varias de las funcionalidades de un automóvil dentro de un sistema embebido.

En la materia Taller de Diseño de PCBs se desarrolló el layout para un transceiver para asistentes electrónicos y teléfonos celulares.

En la Materia de Procesamiento Digital de señales mi proyecto consiste en procesar digitalmente una imagen no sintética.

1.1. Sistemas Embebidos Avanzados

1.1.1 Introducción

Un Sistema Embebido es una combinación de Hardware y Software ya sea con una capacidad fija o programable que es diseñada para una función específica utilizando los recursos mínimos necesarios para llevar a cabo dicha función.

En este proyecto se pretende realizar un conjunto de tareas basadas en el protocolo de comunicaciones CAN utilizando la tarjeta de desarrollo DEMO9S12XEP100 de Freescale. Además, este proyecto toma como base las fases de desarrollo de software las cuales nos ayudaran a realizar un proyecto de calidad. Se hizo un análisis de los niveles de abstracción para dividir las tareas, fue de tres capas (la aplicación, hardware independiente y hardware dependiente).

El reto más grande durante este proyecto fue la programación ya que ninguno de los integrantes del equipo tenía experiencia programando, se optó por investigar y trabajar juntos para que todos entendiéramos.

1.1.2 Antecedentes

A través de un cuestionario y una entrevista con el cliente se definieron las principales características que debe poseer el producto para satisfacer las necesidades del usuario, dichas características fueron consideradas por el cliente y el diseñador como requerimientos y se especifican a continuación:

- Encendido automático de luces.
- Control manual de la intensidad de luz del tablero.
- Protocolo de control remoto para alarma.
- Enviar mensajes por CAN:
 - Posición de la llave.
 - Velocidad del vehículo.
 - Presión de los neumáticos.
- Gateway CAN-SCI.

Encendido automático de luces:

Una vez encendido el vehículo, las luces externas del automóvil deberán encender automáticamente cuando el nivel de luz del ambiente impida al conductor tener buena visibilidad.

Control manual de la intensidad de luz del tablero:

Las luces del tablero se encenderán una vez que las luces externas del vehículo estén encendidas, a partir de ese momento el usuario podrá controlar su intensidad manualmente.

Protocolo de control remoto para alarma:

El vehículo solo podrá encender una vez que la alarma haya sido desactivada y esto se realizará enviando un patrón específico, el cual será el único capaz de desactivar la alarma.

Enviar mensajes por CAN:

Velocidad del vehículo:

Se tomará la lectura de la velocidad del vehículo en un rango de 0 a 100 Km/h y ésta lectura será enviada periódicamente.

Posición de la llave: Dependiendo de la posición de la llave, se enviarán los mensajes:

- OFF
- STOP
- ON
- START

Presión de los neumáticos: Cuando la presión de los neumáticos se encuentre por debajo del 80% de la presión ideal, se enviará un mensaje de alarma.

Gateway CAN-SCI: Esta funcionalidad consiste en enviar mensajes por CAN y enviar los resultados por SCI y viceversa.

1.1.3 Solución Desarrollada

Con los requerimientos por parte del cliente, se realizó la codificación requerida para cada una de las partes de este proyecto, A través de una máquina de estados se controló el sistema y se ejecutaban los estados siguientes de acuerdo a la programación.

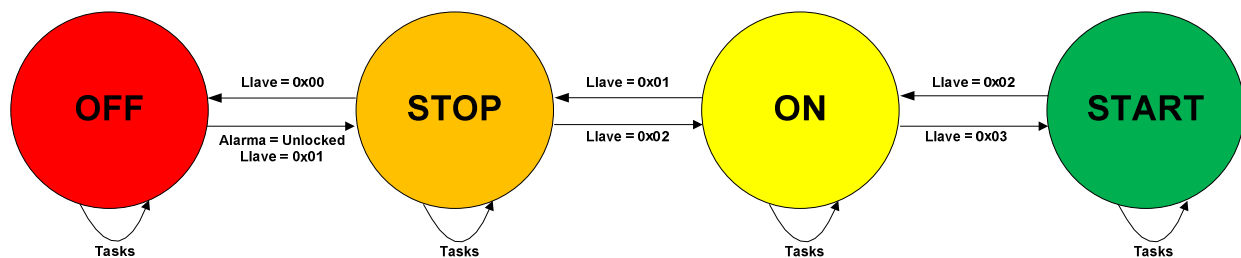


Fig. 1-1 Descripción de la Máquina de estados del Sistema Embebido con representación de la posición de la llave y sus funciones.

Se realizaron pruebas funcionales a cada bloque del diagrama de los niveles de abstracción y una vez que se probó el correcto funcionamiento de cada bloque se realizaron las pruebas con los bloques intercomunicados.

Se conectaron dos *PC's* a una tarjeta DEMO9S12XEP100 cada una, esto por medio del puerto SCI, y se conectaron las tarjetas entre sí, por el puerto CAN como se muestra en la figura 1-2.

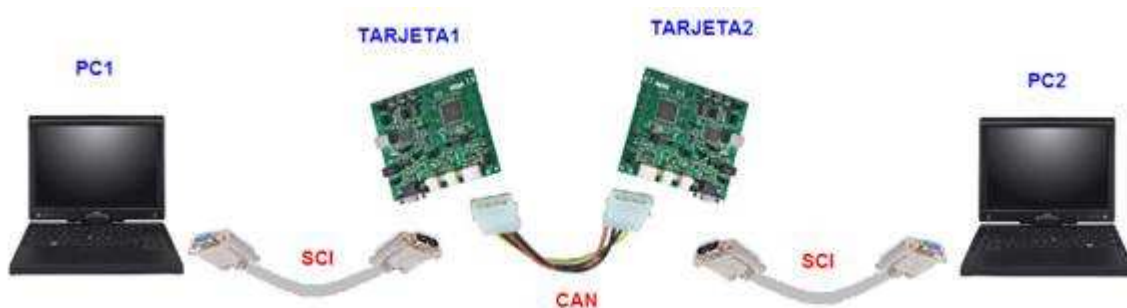


Fig. 1-2 Sistema pruebas basado en dos tarjetas de desarrollo con protocolo de comunicación CAN entre ellas y protocolo SCI hacia el mundo exterior.

1.1.4 Análisis de Resultados

Dentro de nuestro *scheduler* seguimos tres pasos antes de ejecutar las tareas del mismo. Estos pasos consisten en deshabilitar las interrupciones, realizar la inicialización de las variables del sistema y por último habilitar las interrupciones.

Una vez que se han realizado estos pasos es entonces que se realizan las tareas que están dentro del *scheduler*. Existe una interrupción que se ejecuta periódicamente y cada que se ejecuta esta interrupción se determina en qué posición de la llave nos encontramos. Para cada posición de la llave en que nos encontremos corresponderá un estado, teniendo para OFF el estado1, para STOP el estado2, para ON el estado3 y para START el estado4.

Según el estado en que nos encontremos se realizarán diferentes tareas:

Cuando nos encontramos en el estado1 se ejecuta la tarea 1 en la cual verificamos que el código de la alarma sea correcto.

Si estamos en el estado2 es entonces que se realiza la tarea 2, misma que se encarga de mandar por CAN la posición actual de la llave, para este caso “STOP”.

Ahora bien, cuando llegamos al estado3 se pueden realizar las tareas 3, 4, 5, 6 y 7 dependiendo de si ya hemos pasado por el estado4 o aún no. Si aún no hemos pasado por el estado 4 entonces se realiza la tarea 3 en la cual se manda la posición actual de la llave, en este caso “ON”.

Con lo Cual observamos que el sistema respondía correctamente a las acciones ejercidas por Hardware

1.2. Taller de Diseño de PCBs

1.2.1 Introducción

El objetivo de este proyecto es realizar un diseño físico en tarjeta de circuito impreso de un Asistente electrónico para afirmar las prácticas realizadas en el Taller de Diseño de PCBs, se describirá la finalidad de cada una de las prácticas y se mencionen las metodologías utilizadas en cada una de ellas.

Durante el desarrollo del diseño de la tarjeta se pusieron en práctica varios puntos aprendidos durante las clases como fue la creación de símbolos, librerías, la elaboración del esquemático y Layout. El reto más grande fue la elaboración del Layout ya que existen varias reglas que cumplir para la ubicación de los componentes en las diferentes capas de la tarjeta.

1.2.2 Antecedentes

Para realizar las prácticas para llevar a cabo este proyecto fue necesario tener algunos conceptos claros:

Conductive Pattern: Configuración o diseño del material conductor en un material de base como trazos, pads, vías y componentes pasivos (proceso de PCB MFG).

Pad: Porción conductiva para hacer contacto con un componente con un solo pin.

Footprint: Uno o varios pads, todo en conjunto.

Trace: Ruta de acceso conductive en un conductive pattern.

Plated-Through Hole(PTH): Un agujero con placas de metal en sus paredes que hace posible una conexión eléctrica entre los conductive patterns y/o las layers externas.

Vía: Plated-through hole a través de una conexión utilizada como una conexión entre layers pero no hay otros componentes de plomo u otros materiales de refuerzo.

Gerber: Información del diseño en ASCII, coordenadas X y Y de pistas o de pads.

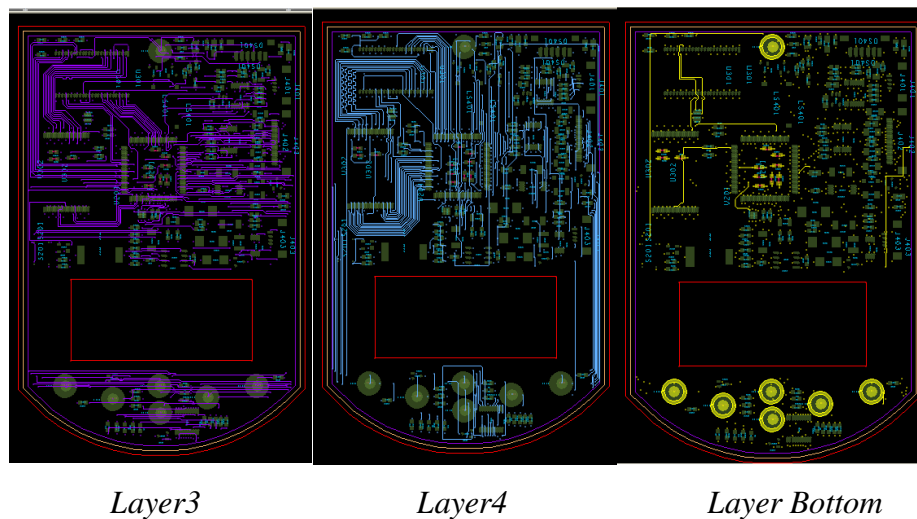
1.2.3 Solución Desarrollada

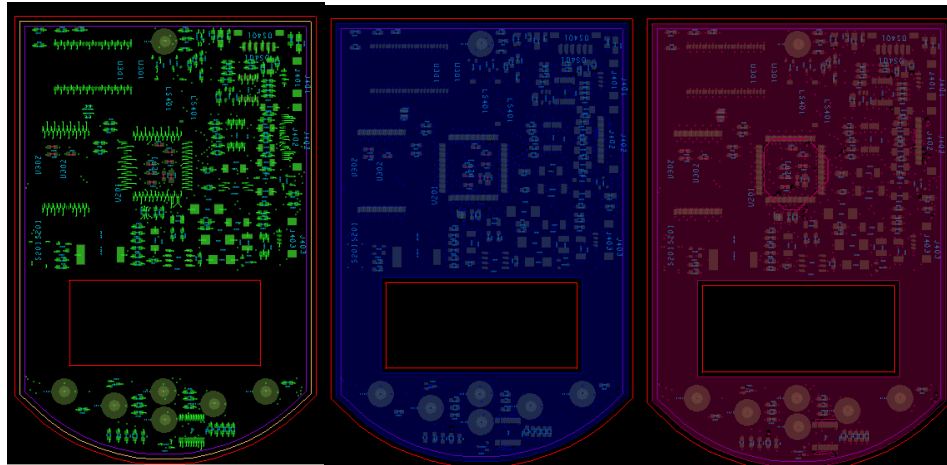
Para realizar el proyecto del diseño de layout de esta palm se utilizó la siguiente metodología:

- Se crearon librerías de símbolos en Allegro.
 - Los circuitos identifican sus componentes por medio de símbolos que están normalizados para que todos puedan interpretarlos, estos símbolos están formados por cuerpo y pines del componente.
 - Los pines (Personal Identification Number), pueden ser de diferentes tipos (entrada, salida, bidireccional, de poder, etc.), dependiendo de la información que transporta para el componente.
- Se crearon librerías de empaquetados en Allegro.
 - Un footprint es la representación física real de un dispositivo, usualmente con la distribución de los pad y perforaciones recomendadas
 - Un footprint debe contener los padstacks que se asignaran a los pines del mismo los cuales debe coincidir con los pines de la representación del símbolo eléctrico

- Se capturo el esquemático.
 - Un diagrama electrónico, también conocido como un esquema eléctrico o esquemático es una representación pictórica de un circuito eléctrico. Muestra los diferentes componentes del circuito de manera simple y con pictogramas uniformes de acuerdo a normas, y las conexiones de poder y de señales entre los dispositivos. El arreglo de los componentes e interconexiones en el esquema generalmente no corresponde a sus ubicaciones físicas en el dispositivo terminado.
- Se diseñó el layout.
 - El layout es la representación física real del circuito eléctrico, el esquema de distribución de los elementos del diseño, generalmente se tiene una especificación de outline y retenciones mecánicas proporcionadas por el cliente las cuales se tienen que tomar en cuenta para el diseño a partir de ese punto se hace el placement de los componentes y la interconexión de sus pines
 - El stack nos define el número de layers con el que el PCB contara, eso nos permite jugar con los planos de power, Ground y los layers de señales

A continuación, se muestran los resultados del layout final:

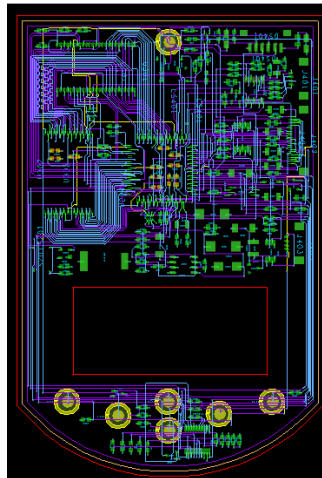




Layer Top

Plano GND

Plano Vcc



Layout final

- Se generaron archivos de fabricación (Gerbers).
 - En esta parte se pretende llevar a cabo la realización de los archivos de fabricación de nuestro circuito usando el diseño de Layout, y revisar la correcta generación de los archivos Gerber para mandarlos a un fabricante y obtener nuestro PCB

1.2.4 Análisis de Resultados

Durante el desarrollo de este proyecto se generaron los archivos de fabricación de nuestro circuito. Para llegar a este punto fue necesario haber concluido previamente el *Layout* del circuito. Se aprendió la manera de generar nuevos *Films* y cómo es que se pueden agregar subclases a los

misimos. Se aprendió cómo generar un *Fabrication Drawing* en el cual se pueden incluir varias especificaciones del *PCB* del circuito.

La realización de los archivos de fabricación fue exitosa pasando a través de todas las etapas necesarias para llegar a este fin en la cual la última parte es la cúspide de todo el esfuerzo de generar librerías y capturas los diagramas esquemáticos y de representación física

1.3. Procesamiento Digital de señales

1.3.1 Introducción

En este documento (Paper) se presenta un algoritmo para la representación no fotorrealista de una imagen, esto es, Se introducirá un algoritmo que produce una imagen que da la impresión de ser una especie de caricatura dibujada a mano de la imagen original, Se presentará un método que divide el problema en categorías distintas, Cuantificación de color, Detección de bordes y Distorsión de bordes.

Esta materia a pesar de que la tome como clase recreativa fue más interesante de lo que esperaba y nos encontramos con retos definir los grupos de colores para la cuantificación de color, sin embargo, hay un tema interesante en este proyecto que fue la transformada de Fourier que con la ayuda de Matlab no fue necesario meternos a hacer cálculos.

1.3.2 Antecedentes

El entretenimiento es uno de los usos más comunes de una computadora en estos días y ha impulsado el crecimiento de la industria informática. Podemos pensar en una larga lista de cosas divertidas que podemos hacer con una computadora, podemos escuchar música, ver una película o jugar a un videojuego, pero hay otro tipo de entretenimiento, uno que no sólo es divertido, sino que también puede ayudar a que las personas se involucren más con el uso de una computadora, la edición digital de fotografía.

Con la fotografía digital es muy fácil de tomar una foto y subirla a una computadora, y una vez que la imagen está en el equipo, puede tener toda la diversión que desee.

Una forma de entretenimiento que puede alcanzar los dos objetivos mencionados anteriormente es la idea de ser capaz de convertir una fotografía personal en una caricatura.

La representación no fotorrealista proporciona una forma de entretenimiento y se ha venido desarrollando en los últimos años, y es muy fácil de encontrar el software que proporciona este tipo de procesamiento de imágenes. en este trabajo se introduce un algoritmo que ofrece un nuevo enfoque para este tipo de procesamiento de imágenes, donde nos centramos en dar un efecto de "hecho a mano" a la imagen que producimos

1.3.3 Solución Desarrollada

Los Algoritmos.

Nuestro reto se resolverá aplicando algoritmos a tres categorías distintas, Cuantización de color, Detección de bordes y Distorsión de bordes

A) Reducción de Color

Para reducir los colores de una imagen se utiliza un proceso llamado de cuantificación, este algoritmo reduce los colores de una imagen a través de tres pasos

1. Formar Grupos con los colores representativos
2. Hacer una paleta con los valores promedio de cada grupo de colores
3. Sustituir los valores de los colores en la imagen original

Para reducir los colores de la imagen original, Primero se hace una reducción de las tonalidades cromáticas

Segundo, cada color tiene un valor en los tres componentes R, G y B, el proceso reduce la imagen original para cada combinación posible de colores para cualquier combinación posible con los valores 51, 102, 153, 204 y 255.

Tercero Definir 22 colores que serán utilizados en la caricatura y agrupar los colores resultantes en el paso 2 con los 22 colores, reemplazar los colores y obtenemos una imagen resultante.

B) Detección de Bordes

En una caricatura es importante distinguir entre diferentes líneas de contorno, Un contorno que tiene más importancia que otro si tiene que dibujar de manera diferente con una línea más gruesa, se obtuvo la transformada indícala de Fourier a través de la función `dwt2` de Matlab para obtener 4 matrices diferentes

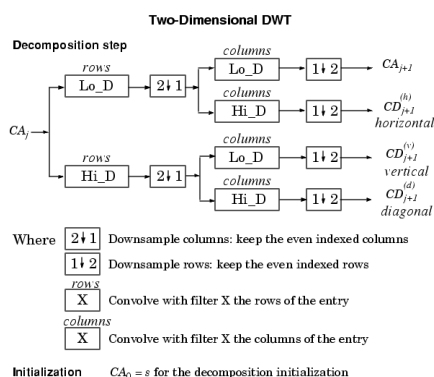


Fig. 1-3 Transformada ondícula en 2 dimensiones

C) Distorsión de Bordes

Cuando la imagen se convierte en caricatura, Para que parezca que ha sido hecha a mano es necesario darles un efecto a los bordes para eso utilizamos un filtro llamado de movimiento.

El filtro de movimiento crea un movimiento que hace que la imagen se vea borrosa, el filtro es capaz de hacer movimientos de manera Radial, Lineal y de acercamiento, cada uno de estos movimientos puede ser ajustado en longitud y ángulo

1.3.4 Análisis de Resultados

Los Resultados del Algoritmo ALFA-TOON únicamente pueden ser evaluados de manera subjetiva, entonces para evaluar el algoritmo se llevó a cabo una encuesta en la cual mostramos una imagen que fue caricaturizada y las personas evaluaron con una calificación promedio de 8 de un rango de 1 a 10

1.4. Conclusiones Generales

Durante la elaboración del proyecto final de la materia de Sistemas Embebidos se aprendió a utilizar el protocolo de comunicaciones CAN y además se comprendieron las fases de desarrollo de software.

Como resultado se lograron implementar exitosamente todas las aplicaciones requeridas para este proyecto, las cuales fueron administradas por un scheduler Round Robin con interrupciones periódicas.

Se presentaron varios problemas con la comunicación serial al utilizar las interrupciones, los cuales fueron resueltos deshabilitando las interrupciones al momento de ejecutar una tarea que requería de dicha comunicación, una vez finalizada dicha tarea, se habilitaron nuevamente las interrupciones.

Con el proyecto de la materia PCBs adquirí experiencia para identificar y llevar a cabo los principales pasos del flujo de diseño de PCBs, ya que durante este proyecto cree el layout de un sistema electrónico implementado en tecnología de tarjeta de circuito impreso, desde la creación de librerías, la captura del esquemático hasta el diseño físico del PCB mismo y la generación de gerbers necesarios para su fabricación.

Aprendí a interconectar los componentes con la ayuda de wires, buses y puertos, también fue necesaria la depuración de errores y se llevó a cabo la generación del netlist del circuito, también me di cuenta de la importancia de hacer un plan de ruteo antes de comenzar con el mismo.

Se aplicó la verificación de reglas de diseño automatizadas en tecnología de circuitos impresos, y se aplicaron reglas recomendadas de layout para manufacturabilidad, ensamble, pruebas y de diseño de layout para minimizar la interferencia electromagnética y otros efectos de alta velocidad, y por último se generaron los archivos necesarios para la fabricación del PCB.

La materia de procesamiento digital de señales me gustó mucho, mi proyecto fue divertido ya que los miembros del equipo teníamos diferentes opiniones sobre el tono asignado a un grupo de colores parecidos. La cuantificación de color, la detección y la detección de bordas completan nuestro algoritmo el cual parece ser una buena opción para el procesamiento digital de imágenes la cual está ganando popularidad entre la gente que le gusta la edición digital de fotografías y con el auge de las redes sociales, lo cual ayuda que sea divertido para el público y acercar más a las personas a la edición digital y al uso de equipo de cómputo y a las Smartphones que puedan contar con este tipo de aplicaciones .

Apéndices

A. SISTEMAS EMBEBIDOS AVANZADOS

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Maestría en Diseño Electrónico

DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA



**SISTEMAS EMBEBIDOS AVANZADOS
Reporte de Proyecto Final**

Proyecto: CAN

PRESENTA:

**Marcial Camacho Brenda María.
Alvarez de la Cruz Liliana Marcela.
González Chavero Hugo Enrique.
González Navarro Martín.
Romero Ríos José Fabián.**

Profesor: Joaquín García Ramírez

INDICE

1. Objetivo	2
2. Introducción	2
3. Requerimientos	7
4. Especificaciones	8
4.1.Diseño	8
4.1.1. <i>Scheduler</i>	8
4.1.2. Niveles de abstracción	11
4.2.Pruebas	13
4.3.Implementación	15
5. Conclusiones	19

1.OBJETIVO

En este proyecto se pretende realizar un conjunto de tareas basadas en el protocolo de comunicaciones CAN utilizando la tarjeta de desarrollo DEMO9S12XEP100 de *Freescale*. Además, este proyecto toma como base las fases de desarrollo de *software* las cuales nos ayudaran a realizar un proyecto de calidad.

2.INTRODUCCION

Controller Area Network – CAN

CAN (*Controller Area Network*) es un protocolo de comunicaciones desarrollado por la firma alemana Robert Bosch GmbH, basado en una topología bus para la transmisión de mensajes en ambientes distribuidos, además ofrece una solución a la gestión de la comunicación entre múltiples *CPUs* (unidades centrales de proceso).

El protocolo de comunicaciones CAN proporciona los siguientes beneficios:

- Es un protocolo de comunicaciones normalizado, con lo que se simplifica y economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red común o bus.
- El procesador anfitrión (host) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas.
- Al ser una red multiplexada, reduce considerablemente el cableado y elimina las conexiones punto a punto, excepto en los enganches.

Aplicaciones de CAN en el área automotriz

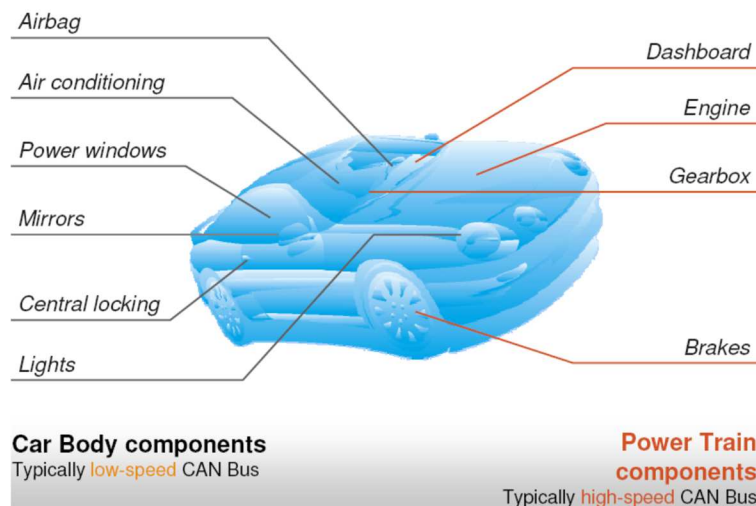


Figura 2.1 Aplicaciones de CAN

CARACTERÍSTICAS:

CAN se basa en el modelo productor/consumidor, el cual es un concepto, o paradigma de comunicaciones de datos, que describe una relación entre un productor y uno o más consumidores. CAN es un protocolo orientado a mensajes, es decir la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, con el cual los nodos deciden aceptar o no dicho mensaje. Dentro de sus principales características se encuentran:

- Prioridad de mensajes.
- Garantía de tiempos de latencia.
- Flexibilidad en la configuración.
- Recepción por multidifusión (*multicast*) con sincronización de tiempos.
- Sistema robusto en cuanto a consistencia de datos.
- Sistema multi-maestro.
- Detección y señalización de errores.
- Retransmisión automática de tramas erróneas
- Distinción entre errores temporales y fallas permanentes de los nodos de la red, y desconexión autónoma de nodos defectuosos.

FRAMES:

Formato Estandar (CAN 2.0A): Identificador de 11 bits.
 2^{11} identificadores posibles

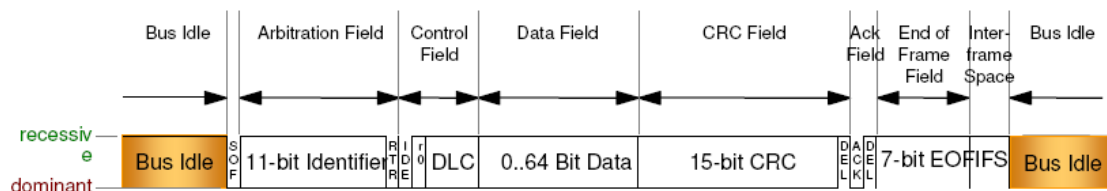


Figura 2.2 Formato Estándar CAN

Breve descripción de los *frames*.

SOF (*Start of Frame*)

- Marca el comienzo de cada *frame* de CAN.
- Es siempre un bit dominante (0).
- Proporciona un flanco de bajada para la sincronización de transmisor y receptor.

Identificador de 11 bits (Campo de arbitraje, *frame* estándar)

- Contiene el identificador (11 bits para CAN 2.0A) el cual es usado para arbitraje.
- El identificador determina la prioridad de los *frames*: bajo identificador=alta prioridad.

- Los siete bits más altos del ID no deben ser todos recesivos (es decir, no deben estar todos en 1).
- El bit RTR (Solicitud de transmisión remota) es siempre dominante en un *Frame* de datos.

Campo de control

- El bit Identificador de Extensión (IDE) es dominante para *frames* estándar.
- El bit r0 no es usado.
- DLC (Código de longitud de datos) es de 4 bits e indica el número de bytes de datos en el campo de datos; puede tomar valores que van de 0 a 8, otros valores no son permitidos.

Campo de Datos

- Contiene la información actual la cual es transmitida.
- El número de bytes de datos puede tomar valores de 0 a 8 en unidades de bytes.
- El número de bytes de datos es dado por el DLC.
- La transmisión empieza con el primer byte de datos (byte 0).

Campo de CRC

- Contiene los 15 bit del código de CRC. Utiliza un generador polinomial $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- CRC es un complejo, pero rápido y efectivo método de detección de errores.
- El Delimitador de campo de CRC (DEL) marca el final del campo de CRC.
- El Delimitador de campo de CRC es siempre recesivo.

Campo de *Acknowledge*

- Contiene el bit de *Acknowledge*.
- El bit de *Acknowledge* puede ser dominante (Tx) o recesivo (Rx)
- El Delimitador de campo de *Acknowledge* (DEL) marca el final del campo de *Acknowledge*.
- El Delimitador de campo de *Acknowledge* es siempre recesivo.

EOF (*End of frame*)

- Consiste de 7 bits recesivos consecutivos.
- Marca el final del *frame* de datos.

RKE (*Remote Keyless Entry*)

Los vehicules están típicamente proveídos de un número de características de seguridad separadas. Un sistema RKE (*Remote Keyless Entry*) permite al usuario bloquear y desbloquear su automóvil a distancia sin utilizar una llave. Un radio receptor de entrada remota es montado en un vehículo el cual responde al transmisor de radio portado por el usuario del vehículo. Para poder acceder al

vehículo, el usuario presiona un botón de desbloqueo de la llave, el cual envía una señal con la información de autenticación y envía un mensaje de desbloqueo del vehículo.

TMPS (Sistema de Monitoreo de Presión de Llantas)



Figura 2.3 Sistema de monitoreo de la presión de las llantas

Un sistema de monitoreo de presión de llantas o TPMS es un sistema electrónico para monitorear la presión del aire en la llanta de un vehículo y transmitir automáticamente una advertencia al conductor en caso de que la llanta esté desinflada y, en algunos casos, sobre inflada. Se ha legislado para que estos sistemas se instalen progresivamente en todo vehículo de transporte de pasajeros nuevo en los EEUU y algo similar está ocurriendo alrededor del mundo.

La mayoría de los sistemas utilizan sensores Directos para llanta, los cuales transmiten información sobre la presión en la llanta al conductor del vehículo en tiempo real, ya sea a través de una pantalla en el tablero de control o por medio de una luz de advertencia conocida como luz de rodada ("tread light" en inglés). Los sensores son transductores de presión que van unidos a la parte trasera del tallo de la válvula o van unidos de alguna manera a la rueda. Los sensores son manufacturados por compañías Internacionales tales como *Schrader*, *Siemens VDO* (ahora *VDO Automotive AG*), *TRW*, *Pacific Industrial*, *Lear* y otras. No hay un estándar técnico común para los sensores y los fabricantes de EO y proveedores han generado una multitud de diseños de sensores.

Los sensores del Sistema de Monitoreo de Presión de Llantas también transmiten datos tales como el identificador (ID) único del sensor, temperatura y presión, carga de la batería (si es aplicable) y otros datos de diagnóstico. Esta información es enviada a la Unidad de Control Electrónico (ECU por sus siglas en inglés) o al receptor específico del vehículo. Aquellos vehículos que despliegan gráficamente la posición del sensor pueden entonces identificar cual rueda tiene un problema.

¿Qué es una herramienta de monitoreo de presión de llantas?

No existe un estándar común para los sensores del Sistema de Monitoreo de Presión de Llantas, habiendo variaciones que incluyen frecuencias, modulaciones, estructura de datos, protocolos de comunicación, ajuste mecánico y muchos otros factores.

Algunas de las características que pueden tener los TPMS incluyen:

Los sensores del TPMS son unidos a la rueda durante el proceso de ensamblaje de la rueda y la llanta.

- Las ruedas son colocadas en el vehículo. Este es el primer momento en el que el TPMS puede ser claramente asociado con el vehículo.
- Antenas fijas de radio son utilizadas para extraer el identificador (ID) único (y otros datos) del TPMS y asociarlos con la posición de las ruedas en el automóvil. La Unidad de Control Electrónico (ECU) puede entonces ser programada con esta información.
- El auto pasa entonces por la Prueba de Rotaciones en donde el sistema es probado.

De manera similar, la agencia automotriz necesita tener disponible una herramienta portátil para el Sistema de Monitoreo de Presión de Llantas, para leer el ID del sensor de la rueda y para programar la ECU del auto. Esto se requiere en caso de que se reemplace el sensor debido a una falla mecánica, electrónica o de la batería o debido al reemplazo de ruedas y sensores por otras ruedas, llantas de invierno, rotación de las llantas, etc. La agencia también tiene que ser capaz de apagar la luz de advertencia del TPMS en el tablero de control.

Sensores de Velocidad

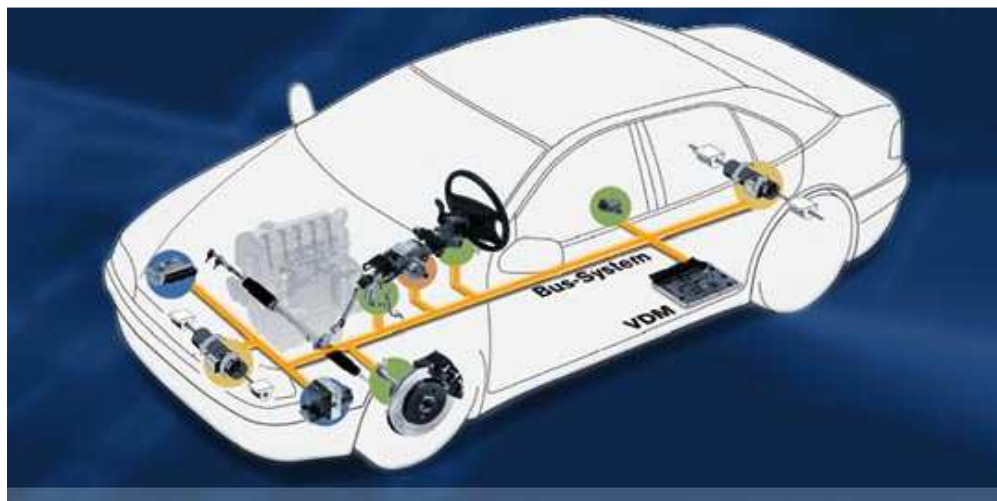


Figura 2.4 Sensores de velocidad

La velocidad que lleva un vehículo o un objeto móvil puede ser calculada mediante un velocímetro. El velocímetro es un dispositivo que mide la velocidad instantánea de un vehículo terrestre.

Hoy en día se usan en todos los automóviles, ya que son indispensables para poder controlar la velocidad del carro, para no excederse a velocidades que se consideran peligrosas. El velocímetro venía como una opción en los modelos de automóviles de 1900 y empezaron a fabricarse como algo estándar después de 1910.

Los velocímetros funcionan de una manera muy sencilla. Los más sencillos están formados por un pequeño imán fijado al cable con una pieza de aluminio con forma de dedal fijada al eje indicador. El imán rota cerca del dedal, provocando cambios en el campo electromagnético. El imán, a su vez arrastra al dedal, así como al indicador, en la dirección de su rotación sin una necesaria conexión entre ellos.

El velocímetro viene muchas veces acompañado de un odómetro y un pequeño interruptor que le envía pulsos a la computadora del carro.

3.REQUERIMIENTOS

A través de un cuestionario y una entrevista con el cliente se definieron las principales características que debe poseer el producto para satisfacer las necesidades del usuario, dichas características fueron consideradas por el cliente y el diseñador como requerimientos y se especifican a continuación:

- Encendido automático de luces.
- Control manual de la intensidad de luz del tablero.
- Protocolo de control remoto para alarma.
- Enviar mensajes por CAN:
 - Posición de la llave.
 - Velocidad del vehículo.
 - Presión de los neumáticos.
- Gateway CAN-SCI.

Encendido automático de luces:

Una vez encendido el vehículo, las luces externas del automóvil deberán encender automáticamente cuando el nivel de luz del ambiente impida al conductor tener buena visibilidad.

Control manual de la intensidad de luz del tablero:

Las luces del tablero se encenderán una vez que las luces externas del vehículo estén encendidas, a partir de ese momento el usuario podrá controlar su intensidad manualmente.

Protocolo de control remoto para alarma:

El vehículo solo podrá encender una vez que la alarma haya sido desactivada y esto se realizará enviando un patrón específico, el cual será el único capaz de desactivar la alarma.

Enviar mensajes por CAN:**Velocidad del vehículo:**

Se tomará la lectura de la velocidad del vehículo en un rango de 0 a 100 Km/h y ésta lectura será enviada periódicamente.

Posición de la llave:

Dependiendo de la posición de la llave, se enviarán los mensajes:

- OFF
- STOP
- ON
- START

Presión de los neumáticos:

Cuando la presión de los neumáticos se encuentre por debajo del 80% de la presión ideal, se enviará un mensaje de alarma.

Gateway CAN-SCI:

Esta funcionalidad consiste en enviar mensajes por CAN y enviar los resultados por SCI y viceversa.

4.ESPECIFICACIONES

En esta sección se describen las etapas de diseño, pruebas e implementación del proyecto.

4.1 DISEÑO

En esta sección se cubrirán los detalles del *scheduler* y los niveles de abstracción.

4.1.1 SCHEDULER

En la figura 4.1.1 se muestra el *scheduler* a utilizar en este proyecto, esta estructura fue elegida tomando en cuenta que nuestro sistema responde principalmente a controles físicos.

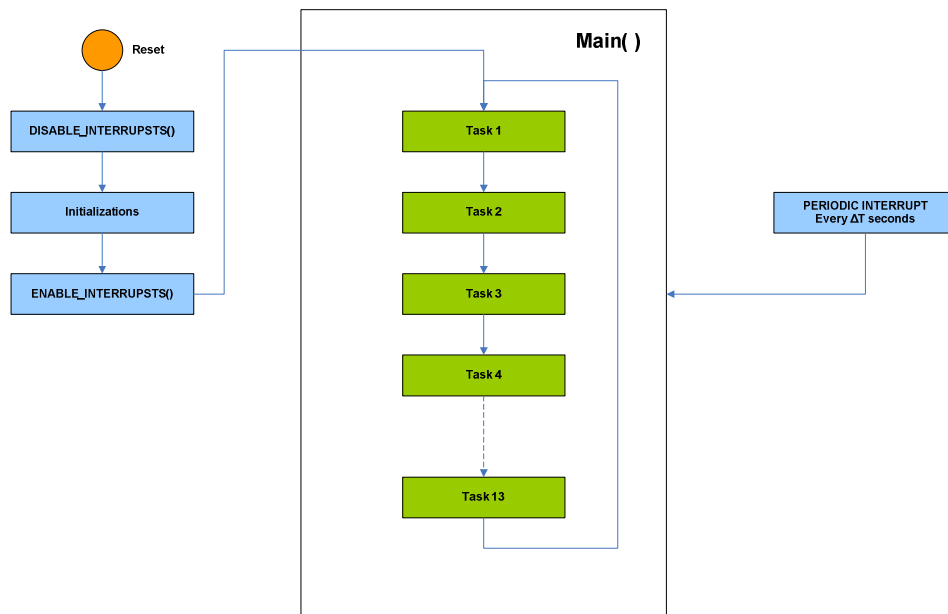


Figura 4,1 Scheduler Round Robin w/time interrupts

Como se puede observar en la figura 4.1 dentro de nuestro *scheduler* seguimos tres pasos antes de ejecutar las tareas del mismo. Estos pasos consisten en deshabilitar las interrupciones, realizar la inicialización de las variables del sistema y por último habilitar las interrupciones.

Una vez que se han realizado estos pasos es entonces que se realizan las tareas que están dentro del *scheduler*. Existe una interrupción que se ejecuta periódicamente y cada que se ejecuta esta interrupción se determina en qué posición de la llave nos encontramos. Para cada posición de la llave en que nos encontremos corresponderá un estado, teniendo para OFF el estado1, para STOP el estado2, para ON el estado3 y para START el estado4.

Según el estado en que nos encontremos se realizarán diferentes tareas las cuales están determinadas en la tabla 4.1 dentro de la columna de Acción.

Estado	Acción
1	OsTask1
2	OsTask2
3	OsTask3
3	OsTask4
3	OsTask5
3	OsTask6
3	OsTask7
4	OsTask8
5	OsTask9
5	OsTask10
5	OsTask11

5	OsTask12
5	OsTask13

Tabla 4.1 Correspondencia entre estados y tareas del scheduler

Cuando nos encontramos en el estado1 se ejecuta la tarea 1 en la cual verificamos que el código de la alarma sea correcto.

Si estamos en el estado2 es entonces que se realiza la tarea 2, misma que se encarga de mandar por CAN la posición actual de la llave, para este caso “STOP”.

Ahora bien, cuando llegamos al estado3 se pueden realizar las tareas 3, 4, 5, 6 y 7 dependiendo de si ya hemos pasado por el estado4 o aún no. Si aún no hemos pasado por el estado 4 entonces se realiza la tarea 3 en la cual se manda la posición actual de la llave, en este caso “ON”.

Si ya hemos pasado por el estado 4 se realizan entonces las tareas 4, 5, 6 y 7. En la tarea 4 se encienden o apagan las luces externas del auto según sea necesario. En la tarea 5 se ajusta la intensidad de la luz interna del auto si y solo si las luces externas están encendidas. Dentro de la Tarea 6 se determina si la presión de las llantas está por encima o por debajo del 80% y se envía por CAN dicho estatus. Finalmente, en la tarea 7 mandamos por CAN el valor de la velocidad del en que se encuentra el auto.

Dentro del estado4 se realiza la tarea 8 en la cual se está enviando únicamente la posición de la llave, en este caso “START”.

El estado5 no pertenece específicamente a alguna posición de la llave ya que este estado fue utilizado para realizar la aplicación de Gateway. Dentro de este estado se ejecutan las tareas 9, 10, 11 y 12. En la tarea 9 se determina por cuál de los buffers (1, 2, 3 o 4) se están recibiendo datos de CAN. Dentro de la tarea 10 se recibe la posición de la llave por el buffer 0 y esta información es enviada por SCI. En la tarea 11 se recibe la información del estatus de la presión de las llantas por el buffer 1 y éste mismo es enviado por SCI. En la tarea 12 se recibe la parte baja de la velocidad del auto por medio del buffer 2 y esta parte baja es enviada por SCI. Finalmente, en la tarea 13 se recibe la parte alta de la velocidad del auto a través del buffer 3 y esta parte alta es también enviada por SCI.

4.1.2 NIVELES DE ABSTRACCION

En esta sección se describen los niveles de abstracción del proyecto, los cuales se muestran en la figura 4.2.

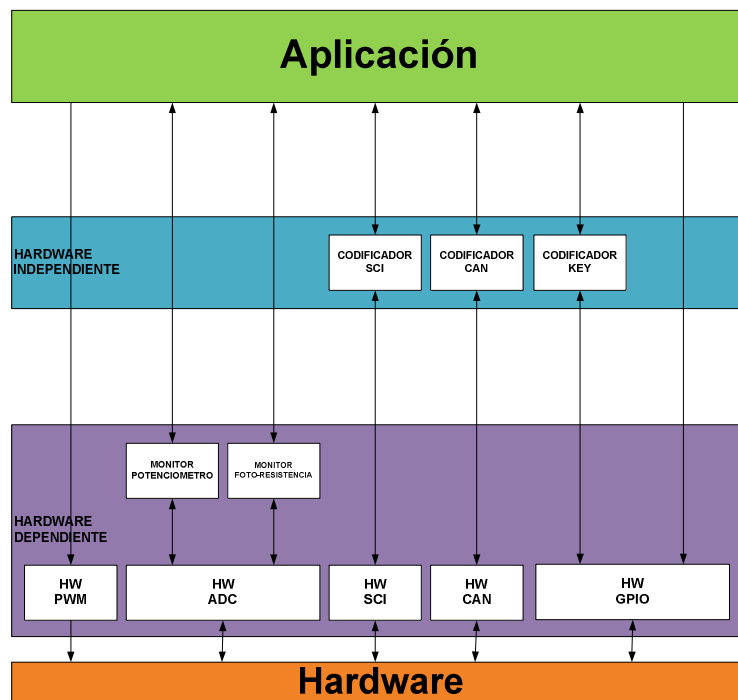


Figura 4.2. Niveles de abstracción

Dichos niveles de abstracción constan de 3 capas, la aplicación, hardware independiente y hardware dependiente.

Aplicación:

En esta capa se determinan las tareas a ejecutar en base a una máquina de estados mostrada en la figura 4.3.

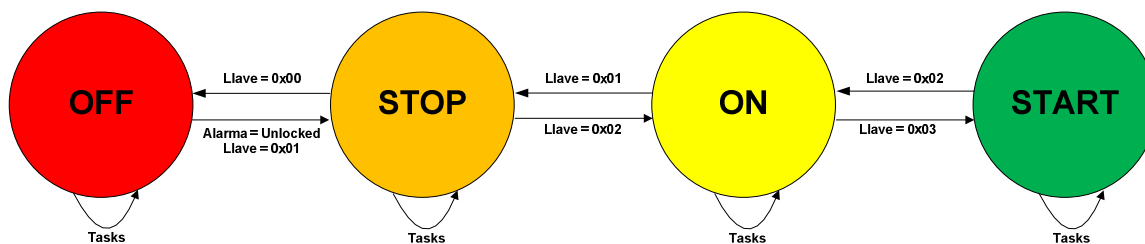


Figura 4.3. Máquina de estados

Antes de comenzar a explicar cada uno de los estados, se explicará la codificación de los mensajes por CAN.

Posición de la llave:

Se envía un paquete de 8 bytes en donde los 4 bytes más significativos son utilizados como un identificador de la posición de la llave 0x0001 y los 4 bytes menos significativos indican la posición de la llave: 0x0010 indican la posición STOP, 0x0100 indica la posición ON, 0x1000 indica la posición START.

Presión de los neumáticos:

Se envía un paquete de 8 bytes en donde los 4 bytes más significativos son utilizados como un identificador de la presión de los neumáticos 0x0100 y los 4 bytes menos significativos indican si la presión está por encima del 80% de la presión 0x1010 o si la presión está por debajo del 80% 0x0101, dicha presión de los neumáticos será suministrada por el potenciómetro de la tarjeta.

Velocidad del vehículo:

La velocidad del vehículo será suministrada por el potenciómetro, en donde estaremos reportando una velocidad de 0 km/h (0x00) a 100 Km/h (0xFF), para enviar esta información por medio de CAN se tomarán los bits menos significativos de la lectura del potenciómetro y se enviará un paquete de 8 bytes en donde los 4 bytes más significativos 0x0010 son utilizados como un identificador de la parte baja de la velocidad mientras que los 4 bytes menos significativos corresponden a la información de la velocidad, de la misma forma se tomarán los bits más significativos de la lectura del potenciómetro y se enviará un paquete de 8 bytes en donde los 4 bytes más significativos 0x0011 son utilizados como un indicador de la parte alta de la velocidad y los 4 bytes restantes indican la información de la velocidad.

Estado OFF:

Espera que se introduzca la clave de la alarma, y solo si esta es correcta se podrá cambiar de estado, de otra forma permanecerá pidiendo el código de la alarma.

Estado STOP:

En este estado se manda solo la posición de la llave por medio de CAN 0x00010010.

Estado ON:

En este estado se ejecutan la mayoría de las tareas, siempre y cuando el estado anterior haya sido START. Las tareas que se ejecutan son: encendido automático de luces, control manual de la intensidad de luz del tablero, la posición de la llave 0x00010100, la presión de las llantas y la velocidad del vehículo.

Estado START:

Se manda la posición de la llave medio de CAN 0x00011000.

Hardware Independiente:

Codificador SCI: La aplicación utiliza este módulo para enviar o recibir mensajes provenientes de la comunicación serial.

Codificador CAN: La aplicación utiliza este módulo para enviar o recibir mensajes provenientes de CAN.

Codificador KEY: Le dice a la aplicación en qué estado se encuentra.

Hardware Dependiente:

Hardware PWM: es el encargado de mandar el dato del ciclo de trabajo el cual sirve para variar la intensidad de la luz.

Hardware ADC: Es el encargado de comunicar a la tarjeta con los sensores analógicos internos y externos.

Los monitores tienen como función principal el chequear cuando el valor anterior del ADC cambia.

Hardware SCI: Este bloque permite a la tarjeta tener una comunicación serial con el exterior, ya sea como receptor o transmisor.

Hardware CAN: Este bloque nos permite por medio del protocolo de comunicación CAN recibir y transmitir mensajes, utilizando 5 buffers de recepción y 3 de transmisión.

Hardware GPIO. La función de este módulo es configurar las entradas y salidas de propósito general.

4.2 PRUEBAS

Se realizaron pruebas funcionales a cada bloque del diagrama de los niveles de abstracción.

Una vez que se probó el correcto funcionamiento de cada bloque se realizaron las pruebas con los bloques intercomunicados.

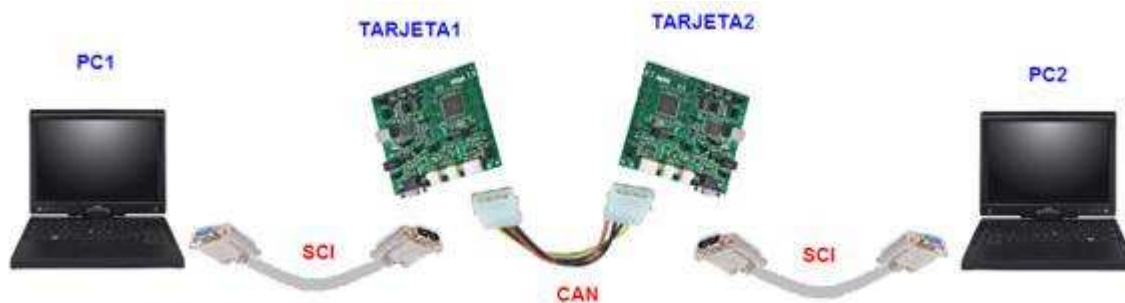


Figura 4.4 Esquema de pruebas funcionales

Se conectaron dos *PC's* a una tarjeta DEMO9S12XEP100 cada una, esto por medio del puerto SCI, y se conectaron las tarjetas entre sí, por el puerto CAN como se muestra en la figura 4.4.

Para realizar las pruebas se realizaron los siguientes pasos:

- El *dip switch* de la tarjeta 1 se coloca en la posición 0x15 (Gateway), para observar los mensajes que se tienen que enviar por CAN en la PC1, y controlar las posiciones del *dip switch* del auto con la tarjeta 2.
- Cuando el *dip switch* está en la posición 0x01 (OFF), le pide el código de la alarma al usuario de la PC2, si éste código no es el correcto vuelve a pedir la alarma y no realizará ninguna tarea, pero si es correcto manda un mensaje de que la alarma es correcta.
- Cuando el *dip switch* está en la posición 0x02 (STOP), la tarjeta2 le manda un mensaje a la tarjeta1 por el puerto CAN, con la posición de la llave y este mensaje será mostrado en la PC1.
- Cuando el *dip switch* está en la posición 0x04 (ON), la tarjeta2 le manda un mensaje a la tarjeta1 por el puerto CAN con la posición de la llave y este mensaje será mostrado en la PC1, si la posición de la llave ya estuvo en la posición 0x08 (START), se podrán realizar otras tareas:
 1. Se enviará la velocidad del vehículo.
 2. Se enviará un mensaje indicando si la presión de las llantas se encuentra por debajo del 80% de la ideal, u otro mensaje si la presión de las llantas está correcta.
 3. Si la fotorresistencia de la tarjeta 2 detecta que el nivel de la intensidad de la luz del ambiente le impide al conductor tener una buena visibilidad, se encenderá un led de la tarjeta 2 emulando ser las luces externas del auto.
 4. Una vez encendidas las luces externas del auto se podrá controlar la intensidad de la luz del tablero del auto por medio del potenciómetro.
- Cuando el *dip switch* está en la posición 0x08 (START), la tarjeta2 le manda un mensaje a la tarjeta1 por el puerto CAN, con la posición de la llave y este mensaje será mostrado en la PC1.

4.3 IMPLEMENTACION

En esta sección se describen las funciones de cada uno de los módulos diseñados para este proyecto.

PWM

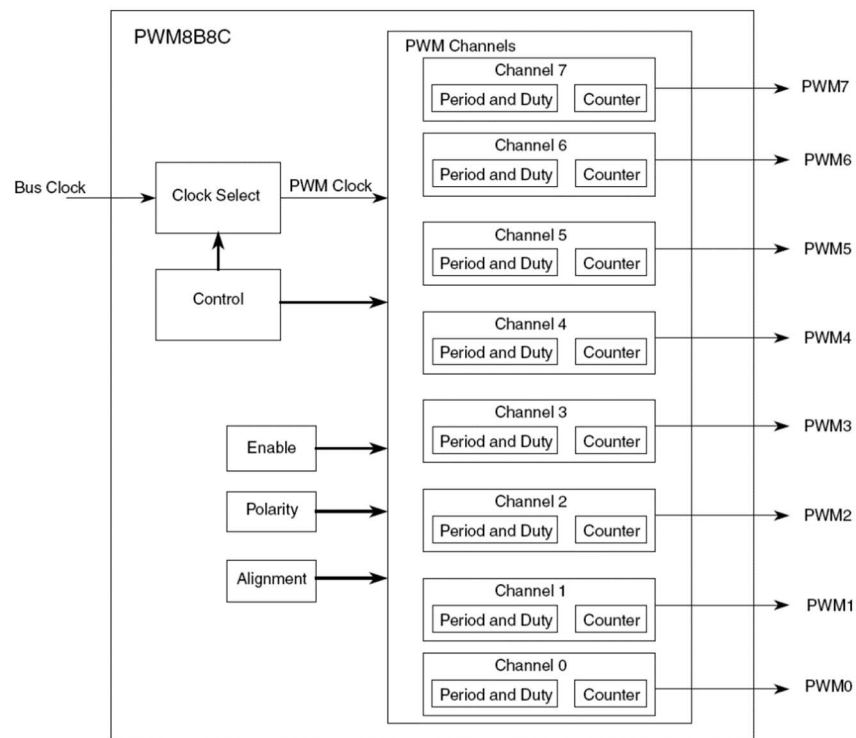


Figura 4.5 Bloque del módulo PWM

Función	<i>void PwmInit(void)</i>
Descripción	Esta función tiene como finalidad inicializar el módulo PWM para el canal 6.

Función	<i>void PWM_SetFrequency(uint8_t SelPrescaler)</i>
Descripción	Esta función establece un número determinado de frecuencias fijas del PWM dependiendo del <i>prescaler</i> .

Función	<i>void PWM_SetDutyCycle(uint8_t valueDutyCycle)</i>
Descripción	Esta función establece el ciclo de trabajo deseado para el PWM.

Función	<i>void PWM_On (void)</i>
Descripción	Esta función habilita al canal 6 como salida del PWM

Función	<i>void PWM_Off (void)</i>
Descripción	Esta función deshabilita al canal 6 como salida del PWM

CAN

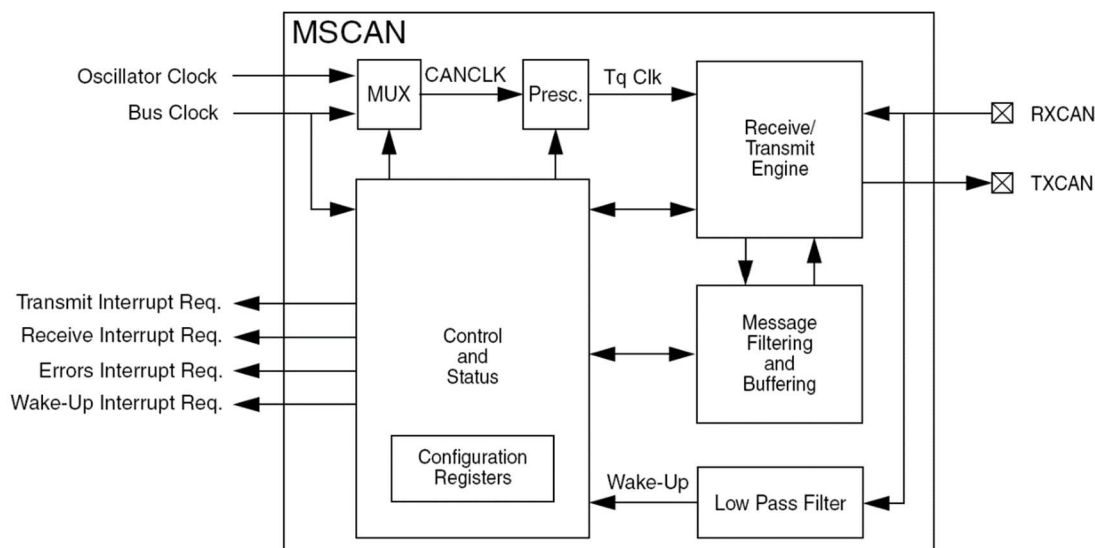


Figura 4.6 Bloque del módulo CAN

Función	<i>void vfnCAN_Init(void)</i>
Descripción	Esta función tiene como finalidad inicializar el módulo de CAN, seleccionar el cristal de la tarjeta como su fuente de reloj, configurar los baudios del protocolo y definir los identificadores de cada uno de los filtros.

Función	<i>void vfnCAN_BaudRateConfig(void)</i>
Descripción	Esta función tiene como finalidad establecer una velocidad de 500kbps con un preescaler de 1.

Función	<i>void vfnCAN_AcceptanceFiltersInit(void)</i>
Descripción	Esta función inicializa los identificadores de cada uno de los filtros de la tarjeta, en este caso, se configuran 4 filtros de 16 bits cada uno.

Función	<i>uint8_t u8CAN_SendFrame(uint32_t u32ID, uint8_t u8Prio, uint8_t u8Length, uint8_t *u8TxData)</i>
----------------	---

Descripción	Esta función tiene como finalidad enviar mensajes por CAN, la cual tiene como parámetros de entrada el identificador, la prioridad, la longitud y el dato, además esta función regresa un estatus de si el mensaje fue enviado o si el mensaje no pudo ser enviado por falta de buffers.
--------------------	--

Función	uint8_t vfnCAN_ReceivedFrame(void)
Descripción	Esta función tiene como finalidad el recibir mensajes por CAN, una vez que un mensaje es recibido, esta función checa a que buffer le corresponde y almacena su información en ella, además esta función regresa un estatus de si existe un nuevo mensaje o no.

SCI

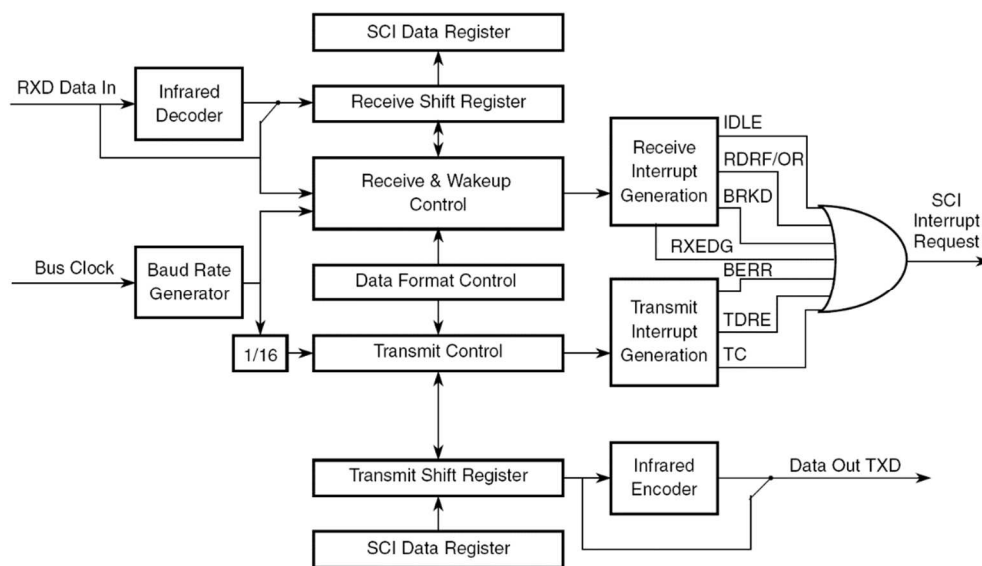


Figura 4.7 Bloque del módulo SCI

Función	void SCI_OpenComm(uint8_t sci_num)
Descripción	Esta función habilita y configura al puerto serial con una velocidad de 57,600 baudios

Función	void SCI_CloseComm(uint8_t sci_num)
Descripción	Esta función deshabilita la comunicación serial.

Función	bool_t SCI_SendBuffer(uint8_t sci_num, uint8_t buffer)
Descripción	Esta función permite enviar datos por el serial.

Función	bool_t SCI_GetBuffer(uint8_t sci_num, uint8_t *buffer)
Descripción	Esta función checa si existe un dato en el buffer de la comunicación serial.

Función	bool_t SCI_CheckGetBuffer(uint8_t sci_num)
Descripción	Esta función permite obtener datos de la comunicación serial.

Función	uint8_t ConvertCharAscii(uint8_t value)
Descripción	Esta función es utilizada para convertir datos en modo ASCII.

Función	void SendString(char *buf)
Descripción	Esta función es utilizada para enviar una cadena de datos por el serial.

Función	void SendHexValue(uint8_t hex_value)
Descripción	Esta función permite enviar valores en hexadecimal por el serial.

ADC

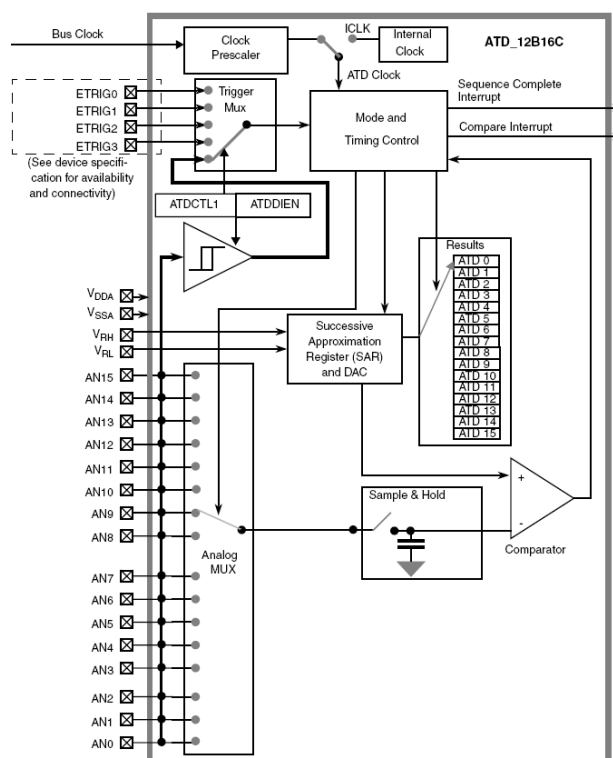


Figura 4.8 Bloque del módulo ADC

Función	<i>void ADC_Init(void)</i>
Descripción	Esta función inicializa al módulo del ADC con una configuración de 8 bits.

Función	<i>void ADC_Task(void)</i>
Descripción	Esta función habilita el registro ATD0 como entrada y permite la múltiple conversión de los registros ATD.

GPIO

Función	<i>void GPIO_Init(void)</i>
Descripción	Esta función configura al puerto A como salida y al puerto B como entrada.

MCU

Función	<i>void Init_XOSC(void)</i>
Descripción	Esta función inicializa al PLL para obtener una frecuencia f_{bus} de 40 MHz

Función	<i>void Init_RTI(uint8_t u8RTI_period)</i>
Descripción	Esta función configura las interrupciones a 5 ms.

5.CONCLUSIONES

Con la elaboración de este proyecto se aprendió a utilizar el protocolo de comunicaciones CAN y además se comprendieron las fases de desarrollo de *software*.

Como resultado se lograron implementar exitosamente todas las aplicaciones requeridas para este proyecto, las cuales fueron administradas por un *scheduler Round Robin* con interrupciones periódicas.

Se presentaron varios problemas con la comunicación serial al utilizar las interrupciones, los cuales fueron resueltos deshabilitando las interrupciones al momento de ejecutar una tarea que requería de dicha comunicación, una vez finalizada dicha tarea, se habilitaron nuevamente las interrupciones.

B. TALLER DE DISEÑO DE TARJETAS DE CIRCUITO IMPRESO

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Maestría en Diseño Electrónico

DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA



TALLER DE DISEÑO DE TARJETAS DE CIRCUITO IMPRESO

Reporte de Proyecto Final

PRESENTA:

Liliana Marcela Alvarez de la Cruz

Profesor: J. Guadalupe Guzmán

Guadalajara, Jalisco. Julio 2010

INDICE

Índice.	2
Objetivos.	3
Introducción.	3
Metodología.	3
Capítulo I Librerías de símbolos.	4
Objetivo.	4
Introducción.	4
Metodología.	5
Procedimiento para generar librerías.	5
Tips.	10
Resultados.	11
Conclusiones.	12
Capítulo II Librerías de empaquetado.	13
Objetivo.	13
Introducción.	13
Metodología.	13
Procedimiento para generar footprints	14
Tips.	18
Resultados.	20
Conclusiones.	21
Capítulo III Captura del esquemático.	22
Objetivo.	22
Introducción.	22
Metodología.	22
Procedimiento para capturar un esquemático y generar su netlist.	23
Tips.	26
Resultados.	27
Conclusiones.	29
Capítulo IV Layout.	30
Objetivo.	30
Introducción.	30
Metodología.	30
Procedimiento para el diseño del layout.	31
Tips.	36
Resultados.	37
Conclusiones.	38
Capítulo V Generación de archivos.	39
Objetivo.	39
Metodología.	39
Procedimiento para generar los archivos de fabricación.	39
Tips.	41
Resultados.	42
Conclusiones.	43
Conclusiones Generales.	43

OBJETIVOS

El objetivo de este documento es realizar un concentrado de las prácticas realizadas en el Taller de Diseño Físico de Circuitos y Sistemas Electrónicos, en donde se describirá la finalidad de cada una de las prácticas y se mencionen las metodologías utilizadas en cada una de ellas.

INTRODUCCION

Para realizar las prácticas para llevar a cabo este proyecto fue necesario tener algunos conceptos claros:

Conductive Pattern: Configuración o diseño del material conductor en un material de base como trazos, pads, vías y componentes pasivos (proceso de PCB MFG)

Pad: Porción conductiva para hacer contacto con un componente con un solo pin.

Footprint: Uno o varios pads, todo en conjunto.

Trace: Ruta de acceso conductive en un conductive pattern.

Plated-Through Hole(PTH): Un agujero con places de metal en sus paredes que hace posible una conexión eléctrica entre los conductive patterns y/o las layers externas.

Vía: Plated-through hole a través de una conexión utilizada como una conexión entre layers pero no hay otros componentes de plomo u otros materiales de refuerzo.

Gerber: Información del diseño en ASCII, coordenadas X y Y de pistas o de pads.

METODOLOGIA

Para realizar el proyecto del diseño de layout de esta palm se utilizo la siguiente metodología:

- Se crearon librerías de símbolos en Allegro.
- Se crearon librerías de empaquetados en Allegro.
- Se capturo el esquemático.
- Se diseño el layout.
- Se generaron archivos.

CAPITULO I. LIBRERIAS DE SIMBOLOS

OBJETIVO

Esta práctica tiene como objetivo elaborar las librerías de 60 símbolos de una lista de componentes (Calvin_BOM_Updated.exe), proporcionada por el maestro, tomando en cuenta sus hojas de especificaciones.

INTRODUCCION

Antes de generar una librería sería bueno saber que una librería es una agrupación de componentes del mismo tipo, con parámetros y características definidas, que pueden ser reutilizadas n número de veces y que cuando se van a crear las librerías no se debe de asumir nada, tenemos que acudir a las especificaciones de los componentes, para esto hay que acudir al dataste del componente, también es importante mencionar que la creación de librerías de símbolos es el primer paso en el flujo de diseño de PCB's.

Un datasheet es un documento que resume el funcionamiento y otras características como una breve descripción funcional, propiedades, esquema de conexiones, medidas, circuito de prueba, información sobre normas de seguridad y uso, etc., de un componente o subsistema con el suficiente detalle para ser utilizado por un ingeniero de diseño y diseñar el componente en un sistema.

Los circuitos identifican sus componentes por medio de símbolos que están normalizados para que todos puedan interpretarlos, estos símbolos están formados por cuerpo y pines del componente.

Los pines (Personal Identification Number), pueden ser de diferentes tipos (entrada, salida, bidireccional, de poder, etc.), dependiendo de la información que transporta para el componente.

El grid es una función que divide un área de trabajo en cuadrículas, y permite integrar dispositivos heterogéneos, por lo que las conexiones entre diferentes dispositivos no generan ningún problema, el grid es escalable.

METODOLOGIA

Para realizar la práctica de la generación de librerías se utilizó la siguiente metodología:

- Se creó un proyecto en Allegro.
- Se dividió la lista de componentes por tipo para agruparlos.
- Se busca la hoja de especificaciones del componente para observar el tipo de pines.
- Se acude al esquemático para observar la distribución de pines de los componentes.
- Se crea el símbolo de los componentes.

PROCEDIMIENTO PARA GENERAR LIBRERIAS

1.- Se abre el programa "Capture CSI", File -> New -> Project..., y crear un nuevo proyecto.

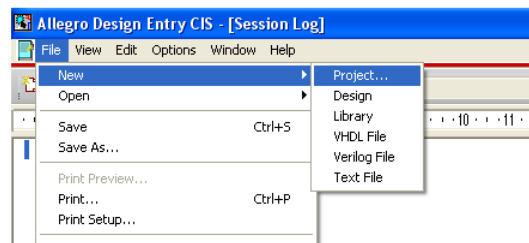


Fig.1 Nuevo proyecto.

2.- Se le da un nombre al proyecto y se especifica la dirección de en donde se guardará.

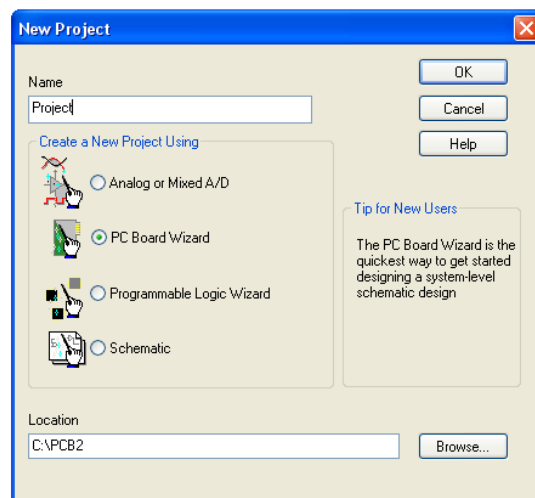


Fig.2 Nombre y dirección del proyecto

3.- Dividimos la lista de componentes por tipo de componente, esto con la finalidad de agrupar librerías por tipo de componente.

	C	D	E	F	G	H
1	Qty	Reference Designator	Manufacturer	Part Number	Pkg	
2	20	C201-C207, C301-C305, C421, C502, C505, C511-C514, C518	AVX	0603YC104KAT2A	C0603	
3	4	C208, C402, C403,	AVX	0805ZC105KAT2A	C0805	
4	2	C209, C210	AVX	06035A200JAT2A	C0603	
5	11	C401, C410-C413, C417, C423-C426, C506	AVX	06035G102ZAT2A	C0603	
6	6	C404, C414, C419, C510, C515, C517	AVX	0603ZC224KAT2A	C0603	
7	5	C405-C408, C508	AVX	12063G225ZAT2A	C1206	
8	2	C409, C509	AVX	08055C105KAT2A	C0805	
9	2	C415, C418	AVX	1206ZG106ZAT2A	C1206	
10	1	C416	AVX	1210ZC103KAT2A	C1210	
11	2	C420, C422	AVX	TAJA226M006RNJ	CA	
12	2	C501, C504	AVX	TAJD107M010RNJ	CD	
13	1	C503	ELNA	DSK-3R3H224U	DSK-621	
14	1	C516	AVX	06035A331JAT2A	C0603	
15	1	D401	Fairchild	BAT54C	SOT-23	
16	2	D402, D403	Fairchild	MMBD914	SOT-23	
17	1	D501	Diodes	S1AB	SMB	
18	1	D502	ON Semiconductor	MBR0540T1	SOD-123	
19	1	DS401	Novalog	MiniSIR2-1	mSIR2	
20	2	J401, J403	Hirose	FH12-4-SA-1SH	FH12-4-1	
21	1	J402	Hirose	FH12-24-S-0.5SH	FH12-24-0.5	
22	3	L401, L501, L502	Panasonic	ELJPB221KF	L1812	
23	1	LS401	Citisound	CHB-03F	CHB-03F	
24	1	Q401	Fairchild	FDN337N	SOT-23	
25	1	Q501	Fairchild	FDV304P	SOT-23	
26	1	Q502	ON Semiconductor	MUN5211T1	SOT-323	
27	1	QF401	Fairchild	FDG6301N	SC70-6	
28	6	R201, R205, R401, R402, R504, R509	Panasonic	ERJ3GEYJ103V	R0603	
29	2	R202, R418	Panasonic	ERJ3GEYJ104V	R0603	
30	2	R203, R506	Panasonic	ERJ3GEYJ331V	R0603	
31	6	R204, R301, R302, R304, R507, R510	Panasonic	ERJ3GEY0R00V	R0603	
32	5	R403, R405, R409, R411, R417	Panasonic	ERJ3GEYJ223V	R0603	
33	4	R404, R406, R408, R410	Panasonic	ERJ3GEYJ330V	R0603	
34	1	R407	Panasonic	ERJ3GEYJ224V	R0603	
35	3	R412-R414	Panasonic	ERJ3GEYJ102V	R0603	
36	1	R415	Panasonic	ERJ3GEYJ334V	R0603	
37	1	R416	Panasonic	ERJ3GEYJ335V	R0603	
38	1	R419	Panasonic	ERJ3GEYJ2R2V	R0603	
39	1	R420	Panasonic	ERJ3GEYJ150V	R0603	
40	1	R421	Panasonic	ERJ3GEYJ2R7V	R0603	
41	1	R501	Panasonic	ERJ3ENF8063V	R0603	
42	1	R502	Panasonic	ERJ3ENF8452V	R0603	
43	1	R503	Panasonic	ERJ3GEYJ515V	R0603	

Fig.3 Lista de componentes.

4.- Creamos librerías, File -> New -> Library, se le da el nombre y la dirección.

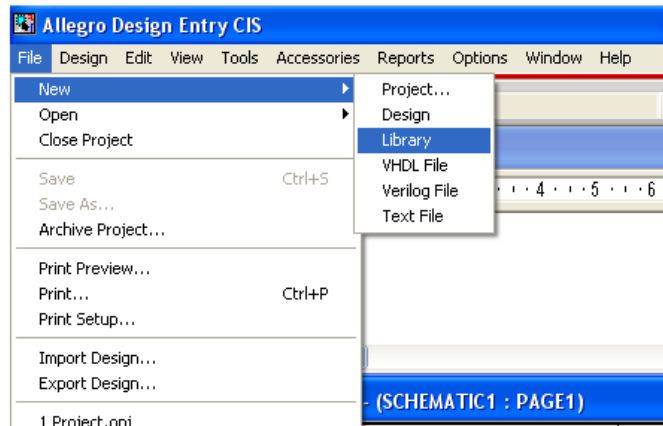


Fig.4 Nueva librería

5.- Seleccionar un componente de la lista (Calvin_BOM_Updated.exe), y acudir a su hoja de especificaciones, para tomar en cuenta la distribución de pines, características del componente y el tipo de pines.

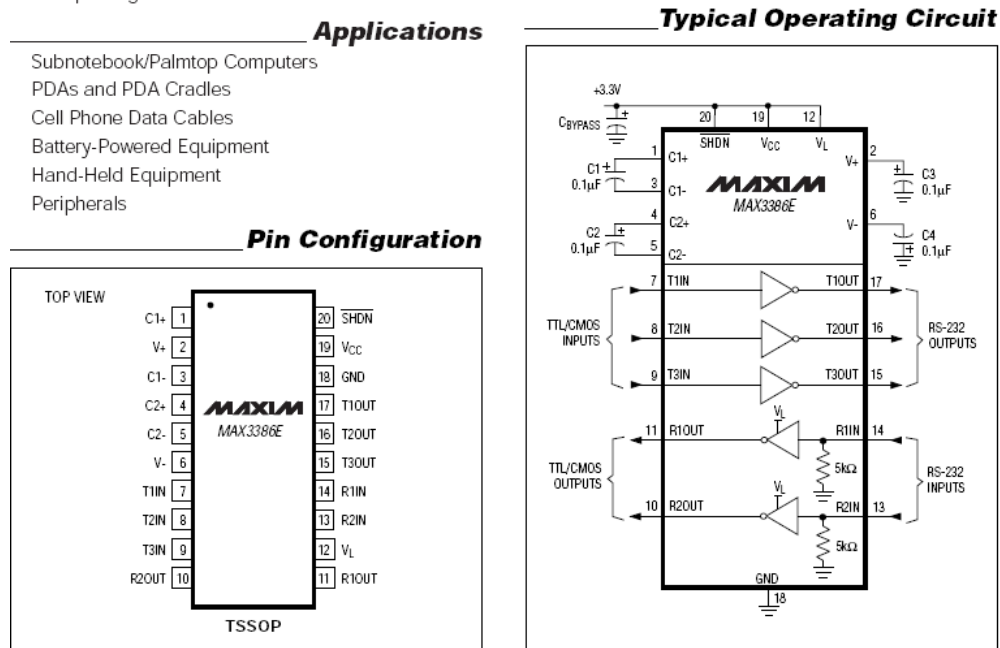


Fig.5 Hoja de especificaciones del MAX3386E.

6.- Acudir al esquemático (Calvin_SCH) que se va a generar con las librerías, para observar la distribución de pines que vamos a necesitar para nuestro proyecto.

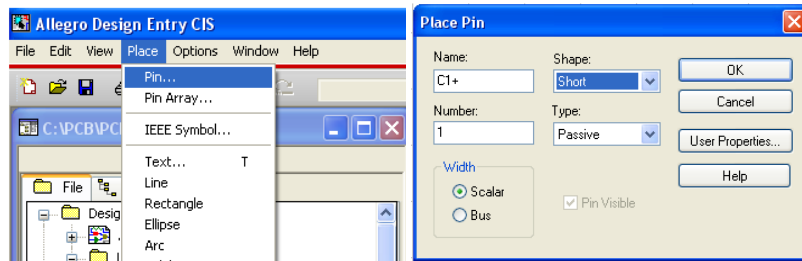


Fig.8 Generar pines.

7.1.2 Se crea el símbolo del componente.

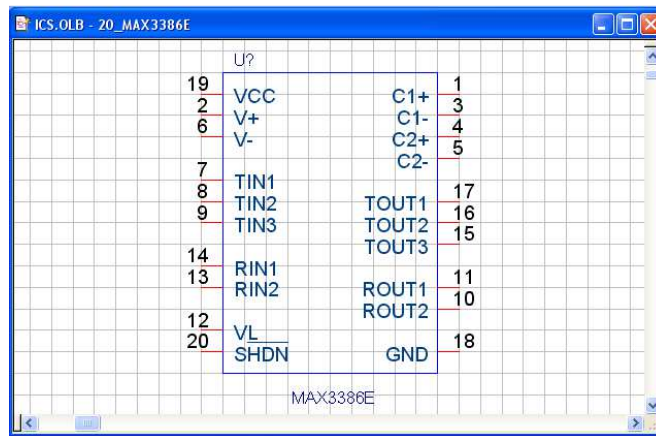


Fig.9 Símbolo del MAX3386E creado desde New Part.

8.2 En la librería seleccionamos "New Part From Spreadsheet" (recomendada para circuitos integrados).

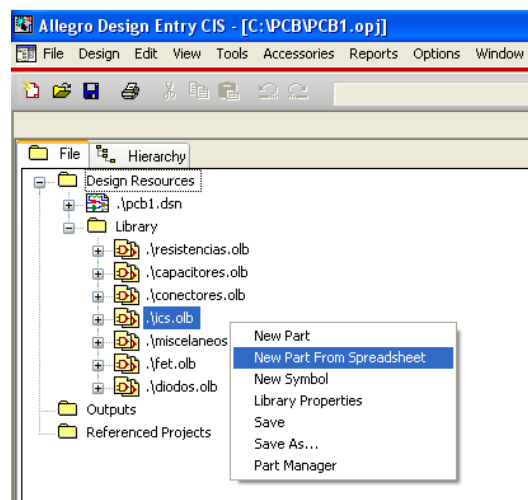


Fig.10 New Part From Spreadsheet.

8.2.1 Se llena la tabla con los datos del componente, si es un circuito integrado con muchos pines se recomienda llenar una hoja de Excel, copiar los datos y pegarlos en la tabla.

Split Part Section Input Spreadsheet

Part Name: 20_MAX3386E
No. of Sections: 1
Part Ref Prefix: U

Part Numbering
Numeric
Alphabetic

	Number	Name	Type	Shape	PinGroup	Position	Section
1	19	VCC	Power	Short		Left	1
2	2	V+	Power	Short		Left	1
3	6	V-	Power	Short		Left	1
4	7	TIN1	Input	Short		Left	1
5	8	TIN2	Input	Short		Left	1
6	9	TIN3	Input	Short		Left	1
7	14	RIN1	Input	Short		Left	1
8	13	RIN2	Input	Short		Left	1
9	12	VL	Power	Short		Left	1
10	20	VS/HIDN	Input	Short		Left	1
11	1	C1+	Passive	Short		Right	1
12	3	C1-	Passive	Short		Right	1
13	4	C2+	Passive	Short		Right	1
14	5	C2-	Passive	Short		Right	1
15	17	TOUT1	Output	Short		Right	1
16	16	TOUT2	Output	Short		Right	1
17	15	TOUT3	Output	Short		Right	1
18	11	ROUT1	Output	Short		Right	1
19	10	ROUT2	Output	Short		Right	1
20	18	GND	Power	Short		Right	1

Add Pins...
Delete Pins
Save As...
Save
Cancel
Help

Fig.11 Tabla a llenar con la opción “New Part From Spreadsheet”

8.2.3 Se puede modificar la figura para adaptar el tamaño con el que genera el componente al tamaño deseado.

9.- Se puede modificar la opción de mostrar o no el numero y nombre de los pines de los componentes, Options -> Part properties, “User Properties”.

Allegro Design Entry CIS - [ICS.OLB - 20_MAX3386E]
File
Edit
View
Place
Options
Window
Help

Preferences...
Design Template...
Autobackup...
Part Properties...
Package Properties...

User Properties
Properties
Name
Value
Attributes
Implementation
Name
20_MAX3386E.Normal
R
Part Reference
U?
R V
Pin Names Rotate
True
Pin Names Visible
True
Pin Numbers Visible
True
Reference
U
R
Value
MAX3386E
V
Pin Names Visible
True
False
True

OK
Cancel
New...
Remove
Display...
Help

Fig.12 Visibilidad de nombre y número de pin.

TIPS

Para evitar problemas futuros se tienen algunos puntos que sería bueno tomar en cuenta durante el proceso de la creación de las librerías de símbolos:

- Mantener siempre activado el snap to grid para hacer más fácil la conexión entre componentes, el cuadrito debe de estar en blanco.
-

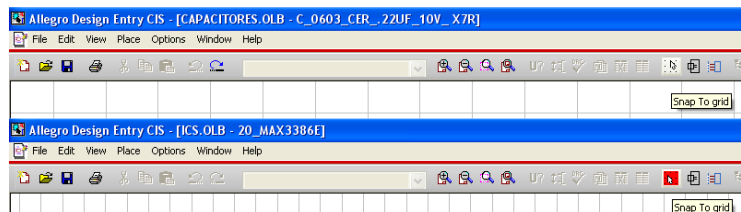


Fig.13 Snap to grid.

- Es importante conservar el Grid spacing para todos los componentes, y esa configuración la puedes encontrar en Options -> Preferences.

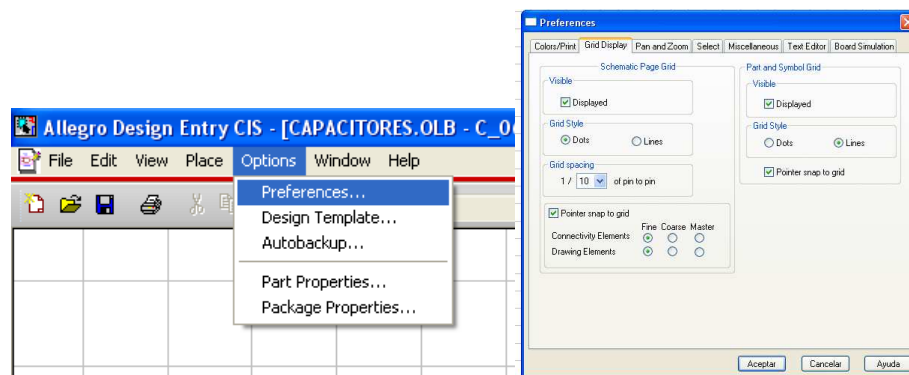


Fig.14 Grid spacing.

- Siempre usar los vértices, cruce del grid, para poner los pines, observando la siguiente figura puedes ver como los pines están sobre el grid, las líneas grises son el grid y la roja el pin.

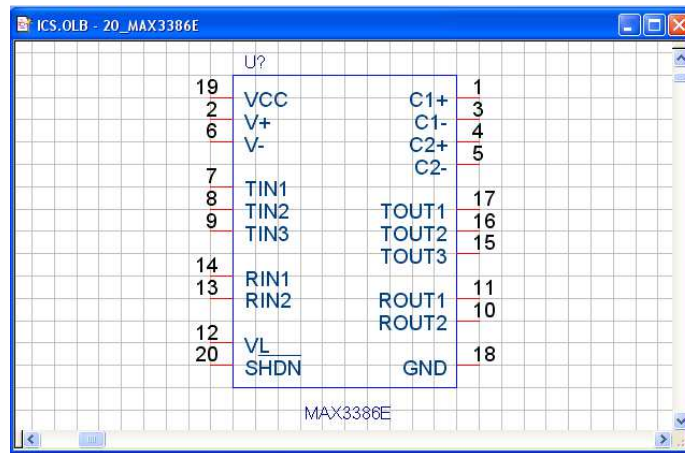


Fig.15 Posición entre grid y pines.

RESULTADOS

Generamos 7 librerías con un total de 59 componentes, un componente más será proporcionado por el profesor:

- Resistencias (20 símbolos)
- Capacitores (13 símbolos)
- Conectores (2 símbolos)
- Ics (11 símbolos)
- Misceláneos (5 símbolos)
- Fet (4 símbolos)
- Diodos (4 símbolos)

CONCLUSIONES

En esta práctica aprendí a generar los símbolos de los componentes que se utilizaran durante el periodo de la materia, para esto se tuvo que tomar en cuenta varios puntos como las especificaciones de los componentes, descritas en su datasheet, y la distribución de pines más cómoda para el proyecto, descrita en el esquemático proporcionado por el maestro.

Se utilizó el programa "Capture CIS" y el procedimiento anteriormente descrito, para generar los símbolos de los componentes.

Esta práctica nos sirvió para jugar con la herramienta y conocerla mejor, con la práctica de generar varios componentes fue más rápida la elaboración de los últimos.

En la elaboración de esta práctica se detectaron varios puntos importantes (tips), y si no los tomamos en cuenta nos puede afectar en las prácticas futuras y corregirlos nos tomaría más tiempo.

CAPITULO II. LIBRERIAS DE EMPAQUETADO

OBJETIVO

Esta práctica tiene como objetivo elaborar los padstack necesarios para generar los footprint de una lista de componentes (Calvin_BOM_Updated.exe), proporcionada por el maestro, tomando en cuenta sus hojas de especificaciones, para esto se podrán tomar en cuenta las medidas de padstack sugeridas en las hojas de especificaciones de los componentes o usar el “**LP Calculador**” para calcular las dimensiones.

INTRODUCCION

Un padstack es la representación de la patita de un componente, los padstacks definen las características de los pines del footprint en cada capa de la placa de un circuito impreso. Entre estas características se encuentran el tipo, el tamaño y la forma, pero es suficiente que se utilice el número y nombre del pin como propiedades, el número de la patita es importante pero el nombre de la misma es solo el aclaratorio para facilitar el conexionado.

Los padstacks pueden ser numéricos o alfanuméricos y pueden situarse en cualquier orden. Los nombres de los pads deben corresponder con los números de los pines (o con los nombres de los pines si no se utilizan números) del símbolo del esquema.

Un footprint es un mapa de las patas de un dispositivo, usualmente con la distribución de los pad y perforaciones recomendadas. La biblioteca permite que los footprint sean dibujados sólo una vez, y luego compartidos por todos los dispositivos de ese tipo. Un footprint es la descripción física de un componente, y el pad es cada patilla del componente que montas en la placa. Antes de crear un footprint se deben definir los padstacks que se asignaran a los pines del mismo.

Un detalle que se debe tener siempre en cuenta y que es causa de errores al crear un pcb, es que el Designator de los pad tiene que coincidir con el del componente en el esquemático.

METODOLOGIA

Para la elaboración de esta práctica se siguieron los siguientes pasos y serán descritos más adelante:

- Se obtiene las medidas de los pads
- Se crean los pads
- Se generan los footprints

PROCEDIMIENTO PARA GENERAR FOOTPRINTS

1.- Obtener las medidas de los diferentes pads que se van a necesitar para generar los footprints.

Para obtener las medidas de los pads se busca la hoja de especificaciones de los componentes, las specs de varios componentes proporcionan footprints recomendados, en esos casos se pueden generar los pads con las medidas recomendadas.

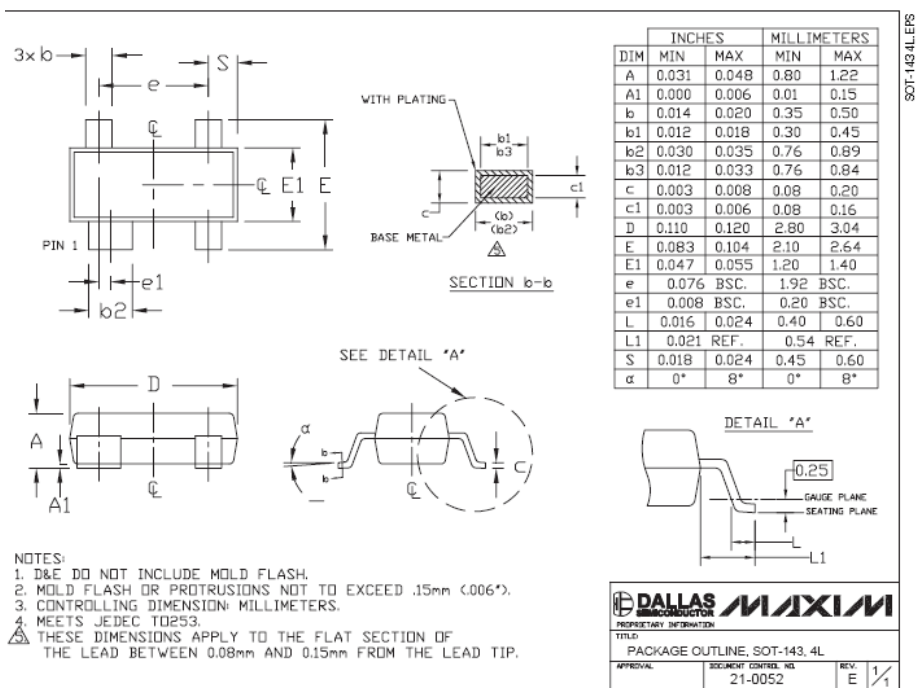


Fig.16 Medidas del empaquetado descritas en spec.

Para crear los pads se utilizo el programa “**Pad Designer**” se llenaron los campos “**Begin Layer, Pastemask Top y Soldermask top**” en base a los datos obtenidos del paso anterior, al campo “**Soldermask top**” se le agregan 6 unidades al valor obtenido.

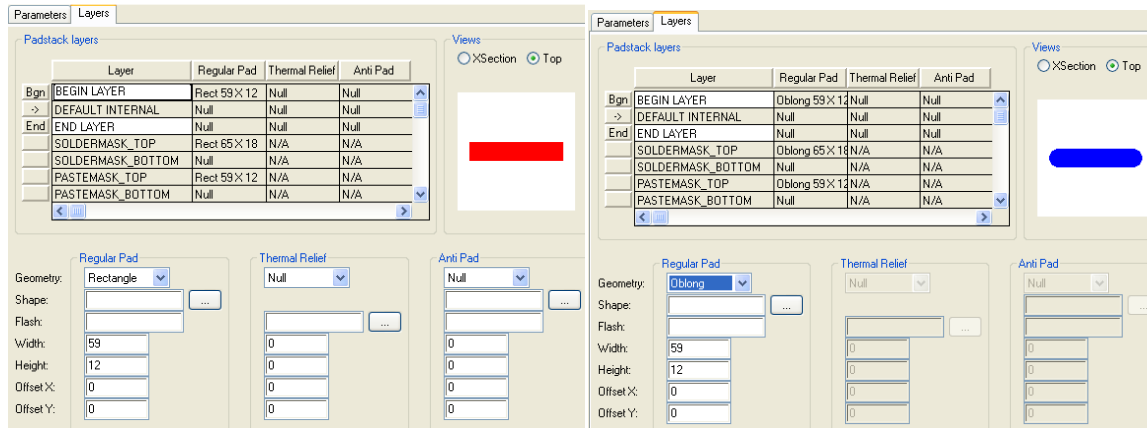
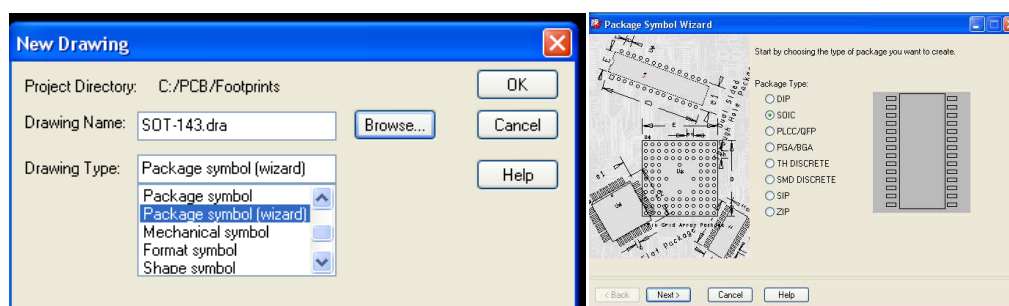


Fig.19 Generación de Pad stack.

En general se crearon pads ovalados pero cuando los pads que se están generando son para un componente que tiene más de 2 pines, se genero un pad rectangular para la representación del pad 1, y para los componentes discretos solo se genero el ovalado.

3.- La generación de los footprints.

Para crear los Footprints se utilizo el programa “**PCB Editor**” en el cual se selecciona el tipo de componente del cual se va a generar el footprint, y utilizando milímetros como unidades de medición e indicando el tipo de componente en el reference designator, se ingresan las mediciones que el programa necesita de acuerdo a los datos obtenidos en el “**LP Calculador**”



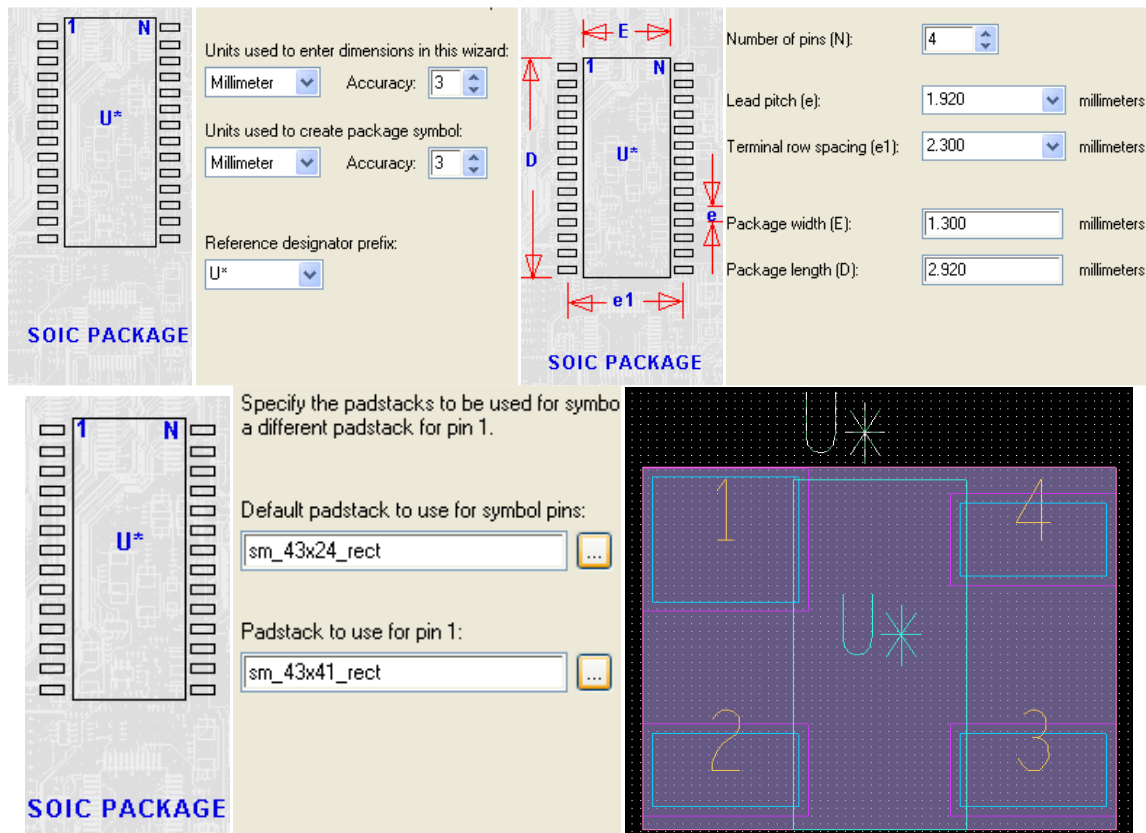


Fig.20 Generación de Footprint.

Se cargan el padstack que va a utilizar el footprint y en caso de ser diferente indicar el padstack para el pin 1, una vez que se obtiene el footprint se pueden hacer las modificaciones necesarias, como eliminar, mover, cambiar número y agregar pines.

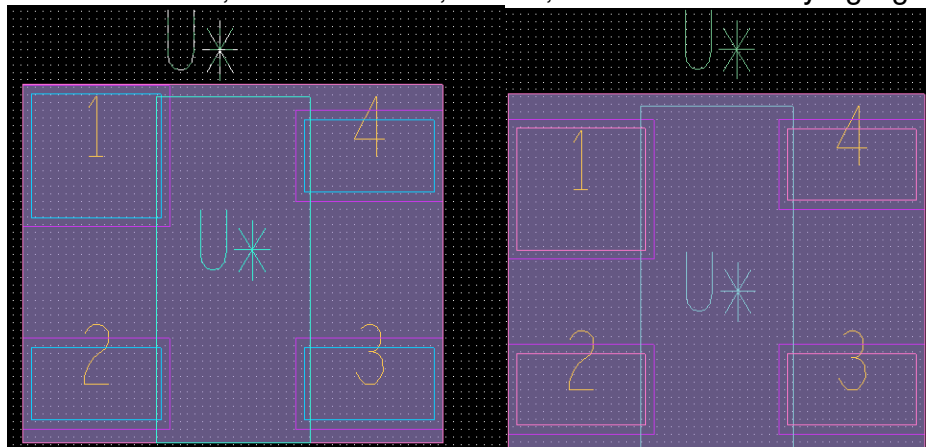


Fig.21 footprint modificado.

Además del pad con diferente geometría para identificar el pin 1 se les coloco un circuito en la parte más cercana al pin 1.

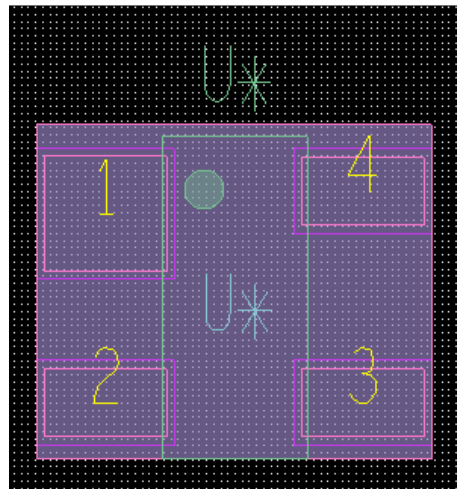


Fig.22 Identificador del pin 1.

A continuación, se muestran algunos ejemplos de footprints:

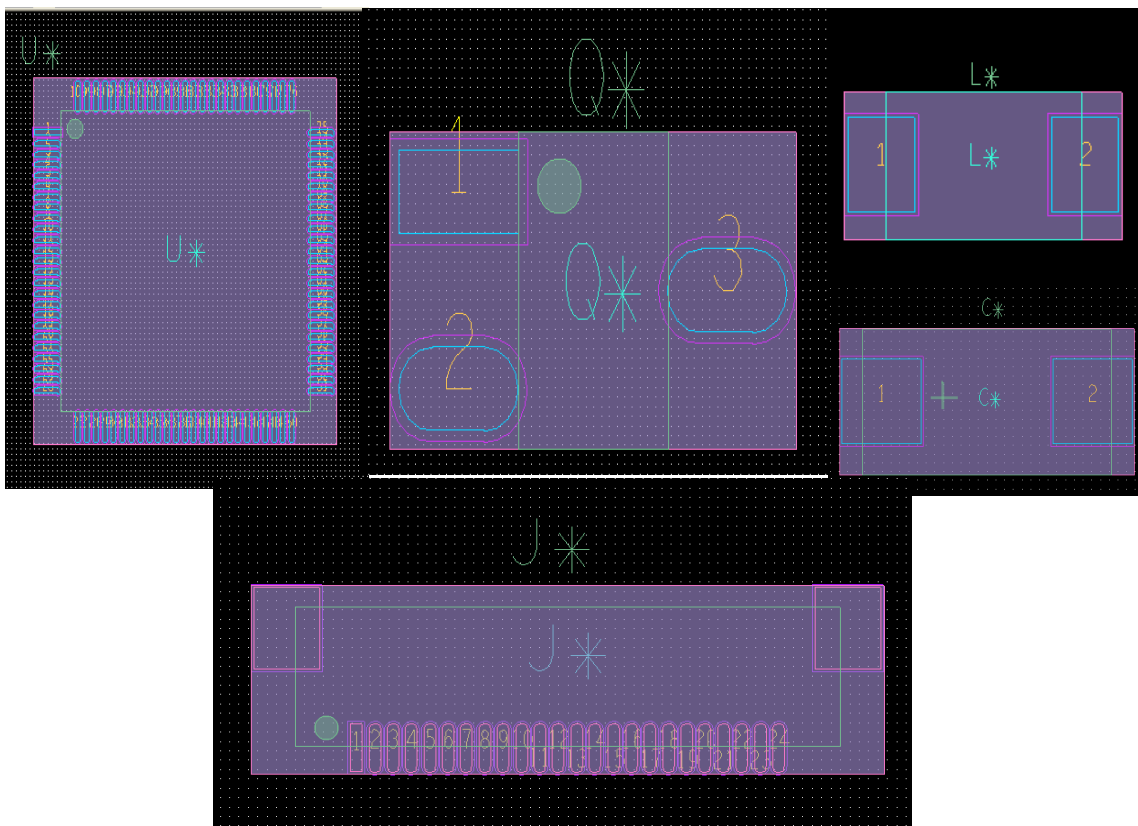


Fig.23 Ejemplos de Footprints.

TIPS

Para evitar problemas futuros se tienen algunos puntos que sería bueno tomar en cuenta durante el proceso de la creación de las librerías de footprints:

- Es importante que el nombre del footprints sea el mismo que se le asigno al símbolo del componente, ya que el nombre es lo que los liga.
-
- Se tiene que definir el pin 1 tomando en cuenta la información de la spec y en los componentes que tienen varios pines poner una marca para identificarlo más fácilmente.

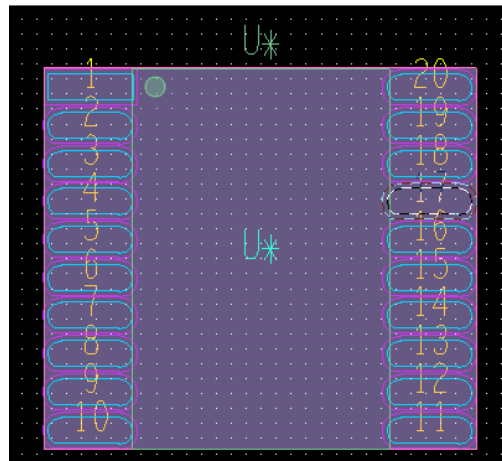


Fig.24 Footprint con identificador del pin 1.

- Tomar en cuenta la imagen que nos muestra “**LP Calculador**” para obtener la información de la longitud y anchura del componente, esto para no tener que mover el dibujo del footprint.

Land Space C	2.30
Land X1	0.60
Land Y	1.10
Land X2	1.05
Silkscreen R1	0.55
Silkscreen R2	2.90
Courtyard V1	3.90
Courtyard V2	3.60

Specify the padstack for a different padstack:

Default padstack: sm_43x24_rect

Padstack to use for:

sm_43x41_rect

Fig.25 Mejor posición del pad para la generación del footprint.

RESULTADOS

Generamos 61 pads que se utilizaron para la generación de 60 footprints de los componentes, el footprint de un componente más (snapdome) fue proporcionado por el profesor, terminando con un total de 61 footprints:

- rectangulares (43 pads)
- Ovalados (18 pads)

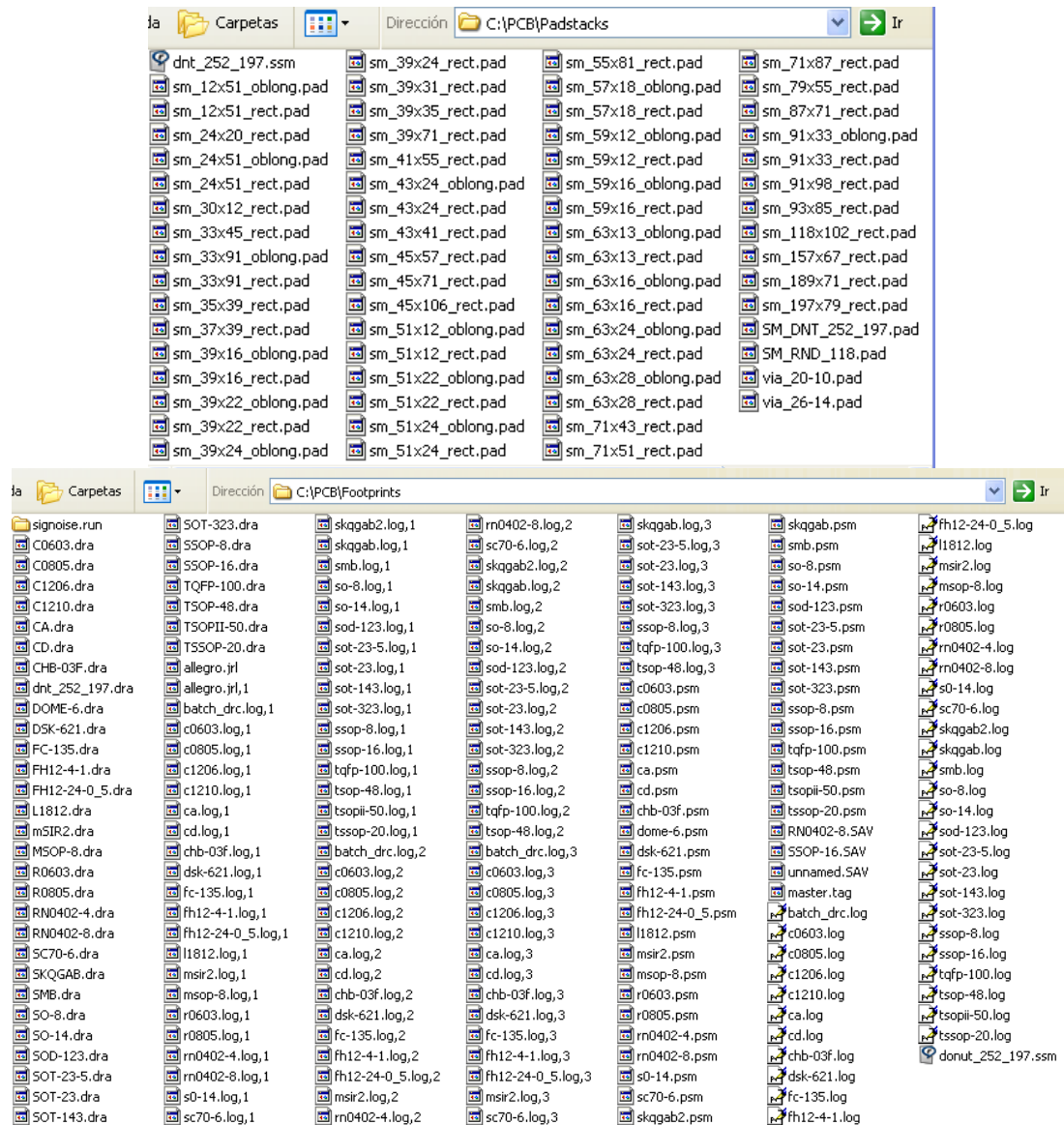


Fig.26 Pads y footprints generados.

CONCLUSIONES

Se utilizaron varias cosas para llevar a cabo esta práctica, se hicieron análisis de las hojas de especificaciones de cada uno de los componentes, y se utilizaron varias herramientas para su elaboración:

- **“LP Calculator”** para obtener las medidas de los pads.
- **“Pad Designer”** para la creación de los pads.
- **“PCB Editor”** para la generación de los footprints de los componentes.

En esta práctica se tomo en cuenta la distribución de pines que proporcionan las hojas de especificaciones de los componentes.

En la elaboración de esta práctica se detectaron varios puntos importantes (tips), y si no los tomamos en cuenta nos puede afectar en las prácticas futuras y corregirlos nos tomaría más tiempo.

La mayoría de footprints fueron generados con la metodología descrita anteriormente y en general fue sencillo obtenerlos, sin embargo se tuvieron dudas en ciertos componentes, como los conectores, pero al final nos dimos cuenta de que la hoja de especificaciones ya propone ciertos footprints.

CAPITULO III. CAPTURA DEL ESQUEMATICO

OBJETIVO

Esta práctica tiene como objetivo realizar la captura de un esquemático en Allegro Design Entry tomando como referencia el archivo Calvin_SCH proporcionado por el maestro y nombrando todas las nets, los buses y los conectores, partiendo del esquemático realizado, compilarlo, depurarlo y generar el netlist del mismo utilizando Allegro PCB Editor..

INTRODUCCION

Un diagrama electrónico, también conocido como un esquema eléctrico o esquemático es una representación pictórica de un circuito eléctrico. Muestra los diferentes componentes del circuito de manera simple y con pictogramas uniformes de acuerdo a normas, y las conexiones de poder y de señales entre los dispositivos. El arreglo de los componentes e interconexiones en el esquema generalmente no corresponde a sus ubicaciones físicas en el dispositivo terminado.

A diferencia de un esquema de diagrama de bloques o disposición, un esquema de circuito muestra la conexión real mediante cables entre los dispositivos.

En un esquemático, los componentes se identifican mediante un descriptor o referencia que se imprime en la lista de partes. Por ejemplo, C1 es el primer condensador o capacitor, L1 es el primer inductor, Q1 es el primer transistor, y R1 es el primer resistor o resistencia. A menudo el valor del componente se pone en el esquemático al lado del símbolo de la parte, pero más detalles adicionales (ocultos) se pudieran enviar e imprimir en la lista de partes. Las leyendas (como referencia y valor) no deben ser cruzadas o invadidas por cables o alambres ya que esto hace que no se entiendan dichas leyendas.

El esquemático se convierte en una *lista de nodos* (o *net list* en inglés). La lista de nodos es una lista de las patas y nodos del circuito, a los que se conectan las patas de los componentes. Usualmente el programa de *captura de esquemáticos*, utilizado por el diseñador del circuito, es responsable de la generación de la *lista de nodos*, y esta lista es posteriormente importada en el programa de ruteo.

METODOLOGIA

Para la elaboración de esta práctica se siguieron los siguientes pasos y serán descritos más adelante:

- Se dividió el esquemático en secciones
- Inserción e Interconexión de símbolos
- Se realizo una revisión de reglas de diseño
- Se genero el netlist del esquemático

PROCEDIMIENTO PARA GENERAR UN ESQUEMATICO Y SU NETLIST:

1.- División del esquemático por secciones.

Debido a que el esquemático que se va a capturar es muy grande es necesario hacer una división del mismo por secciones, para esto se tomo como referencia el esquemático proporcionado por el maestro (Calvin_SCH).

Antes de comenzar con la inserción de los componentes, se definió el tamaño de las páginas que se iban a utilizar para capturar el esquemático.

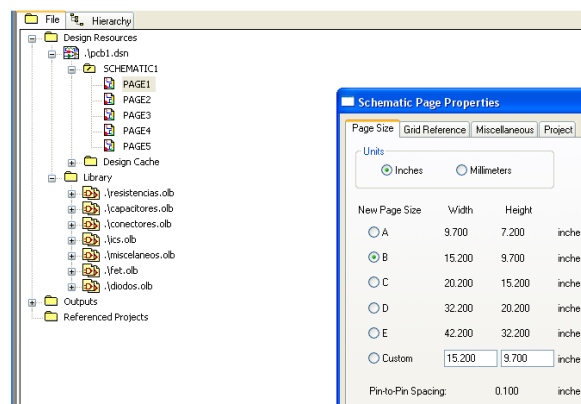


Fig.27 Tamaño de página

2.- Se insertan e interconectan símbolos.

Se agregan las librerías de los símbolos de los componentes, se insertan los componentes que se van a necesitar para la parte del esquemático de cada página respectivamente para posteriormente ser interconectados por medio de wires, a menos que sea necesaria una interconexión con buses.

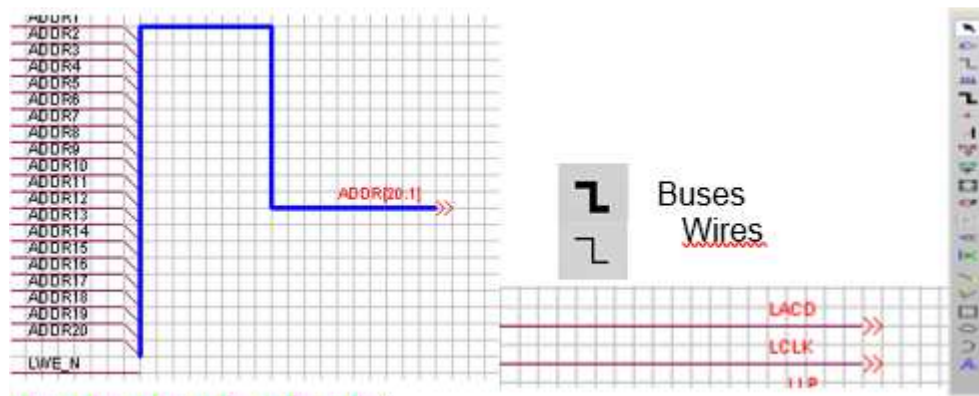


Fig.28 Tipos de interconexiones.

Se nombran todas las nets y buses del esquemático para poder hacer conexiones entre componentes que se encuentran ubicados en diferentes páginas fue necesario utilizar puertos y darles el mismo nombre en las diferentes páginas en las que sería utilizado, y para las fuentes de poder se utilizó el mismo símbolo en todas las páginas.

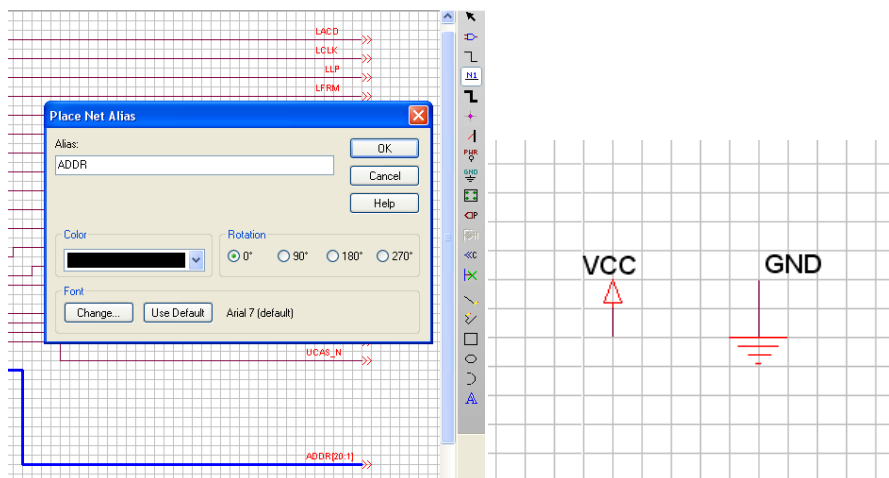


Fig.29 Nombramiento de nets y símbolos de fuentes de poder.

Después de hacer la interconexión de todos los componentes se obtuvo un esquemático de 5 páginas.

3.- Revisión de reglas de diseño.

Una vez terminada la captura del esquemático se llevo a cabo un análisis de reglas de diseño para checar que no tuviéramos errores, esto se hizo con la ayuda de la herramienta en la opción “Design Rules Check”.

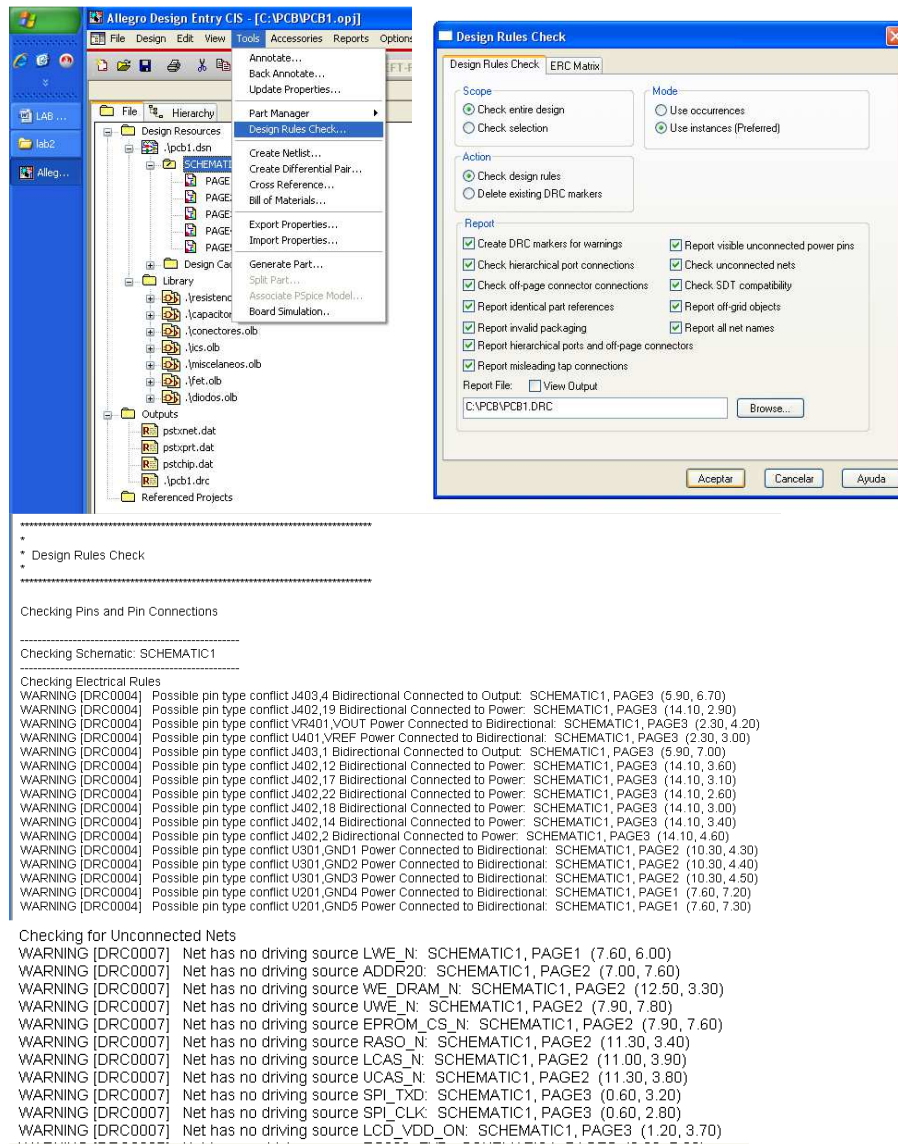


Fig.30 Design Rules Check.

Al encontrarse errores el esquemático fue depurado hasta eliminarlos todos.

4.-Generacion del netlist del esquemático.

Después de que se eliminaron los errores de acuerdo a las reglas de diseño se prosiguió a generar el netlist con la opción “Create Netlist”, al igual que revisando las reglas de diseño durante la creación del netlist se pueden presentar errores y estos deben de ser eliminados.

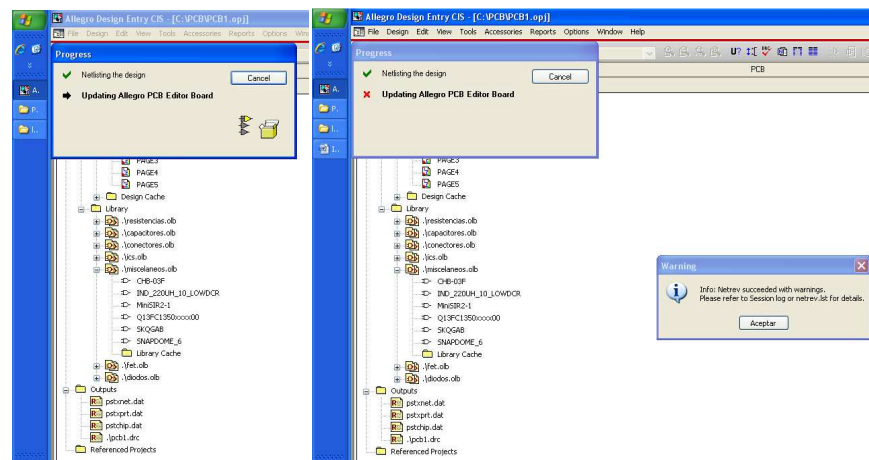


Fig.31 Create Netlist.

TIPS

Para evitar problemas futuros se tienen algunos puntos que sería bueno tomar en cuenta durante el proceso de la captura del esquemático y la creación del netlist del circuito:

- A pesar de que en cualquier momento se puede cambiar el tamaño de página, desde el principio se debe de tomar en cuenta el tamaño.
- Una vez empezado la captura del esquemático, si se realiza algún cambio en el símbolo de algún componente se debe de actualizar la cache para que se refresque el esquemático.

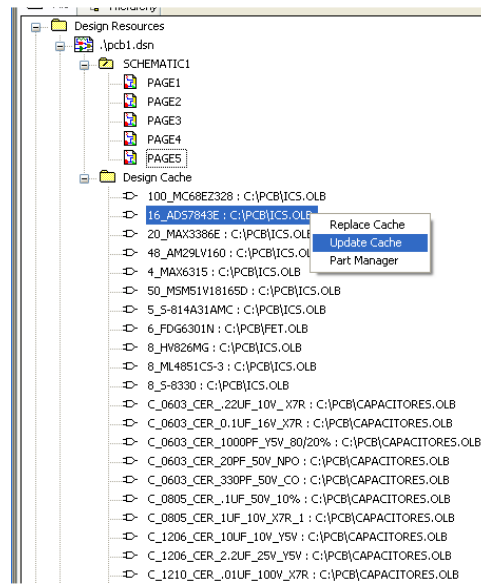


Fig.32 Actualización de la Cache.

- Cuando el pin de un componente no va a ir conectado se le tiene que indicar en el esquemático con el botón de no conectar.

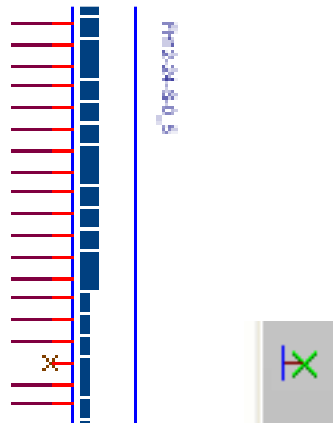
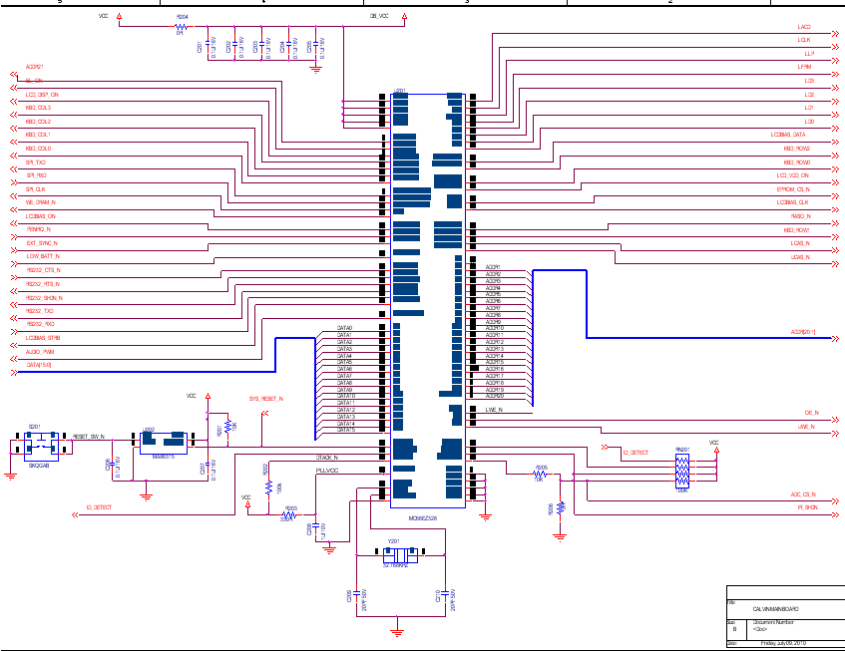


Fig.33 Componente sin conexión.

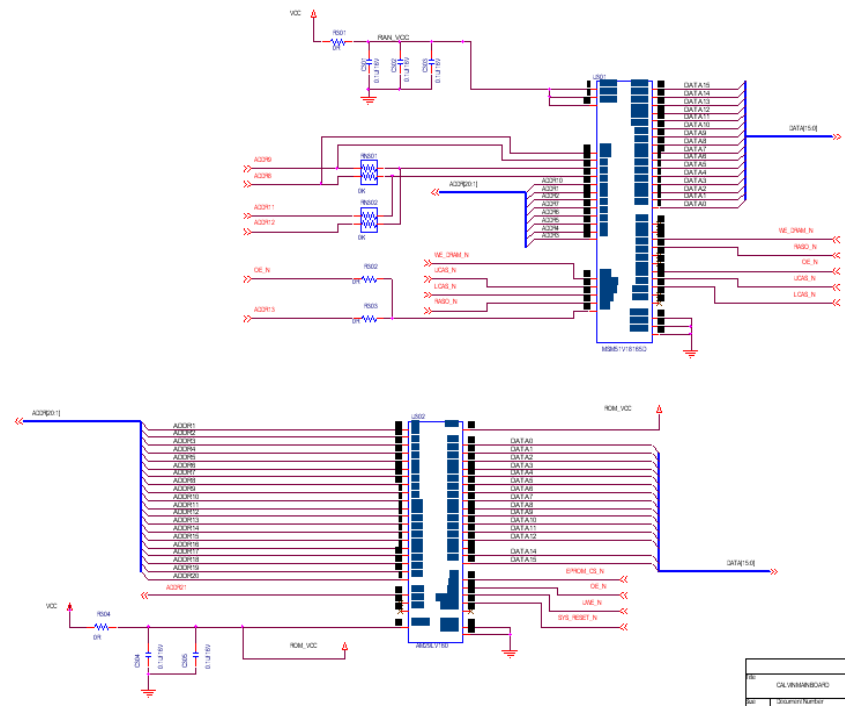
RESULTADOS

El esquemático fue dividido en 5 páginas:

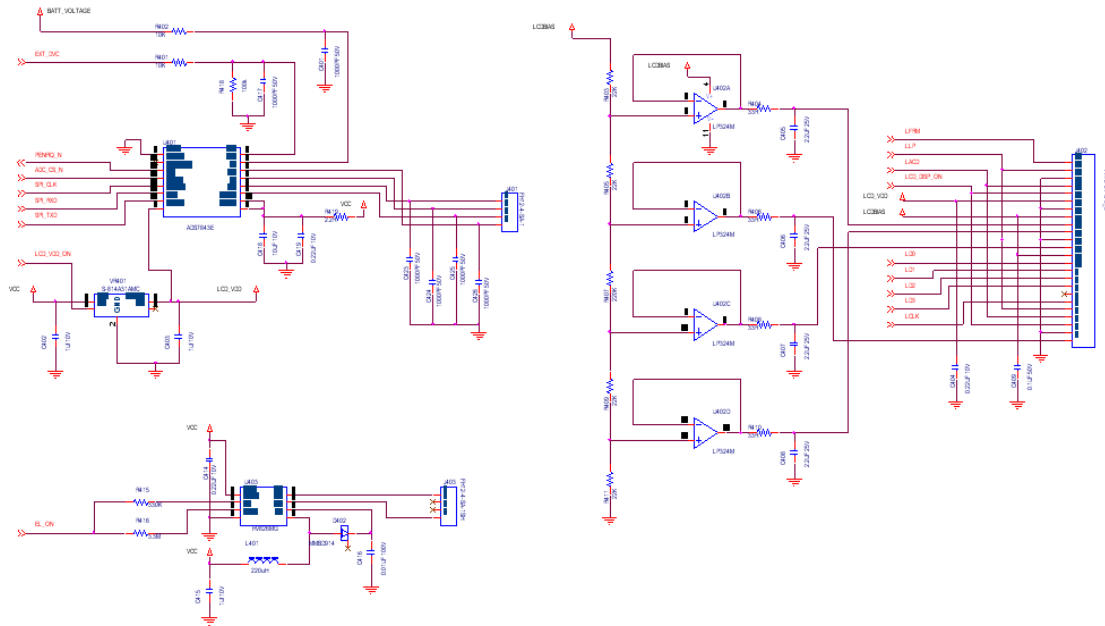
PAGINA 1



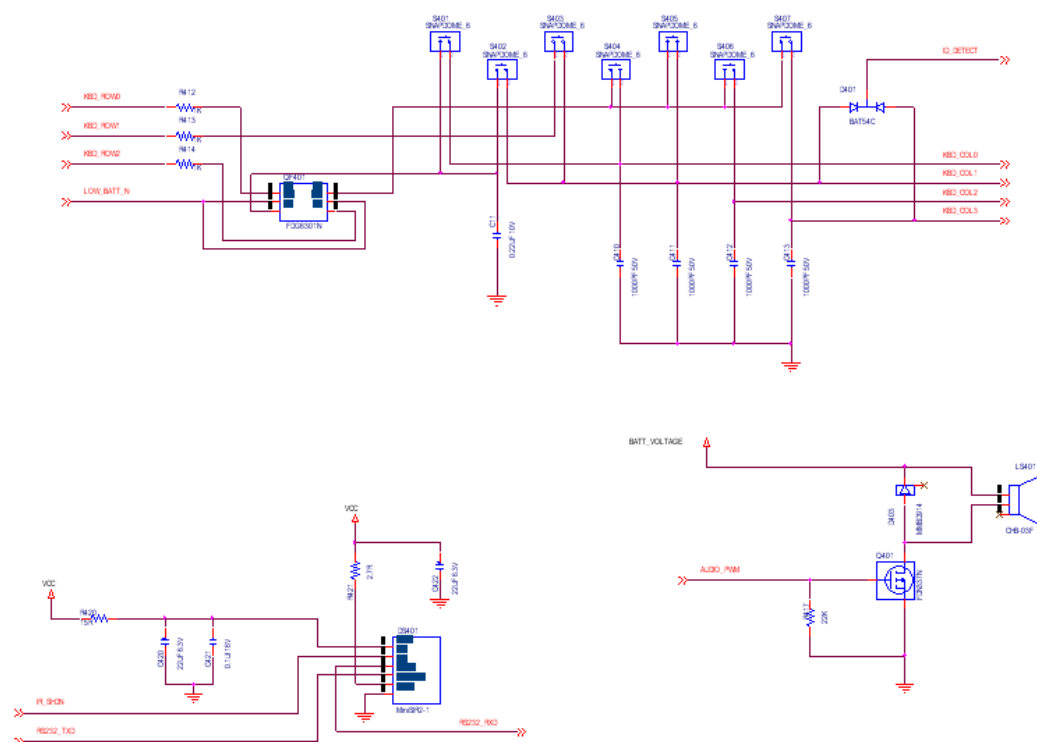
PAGINA 2



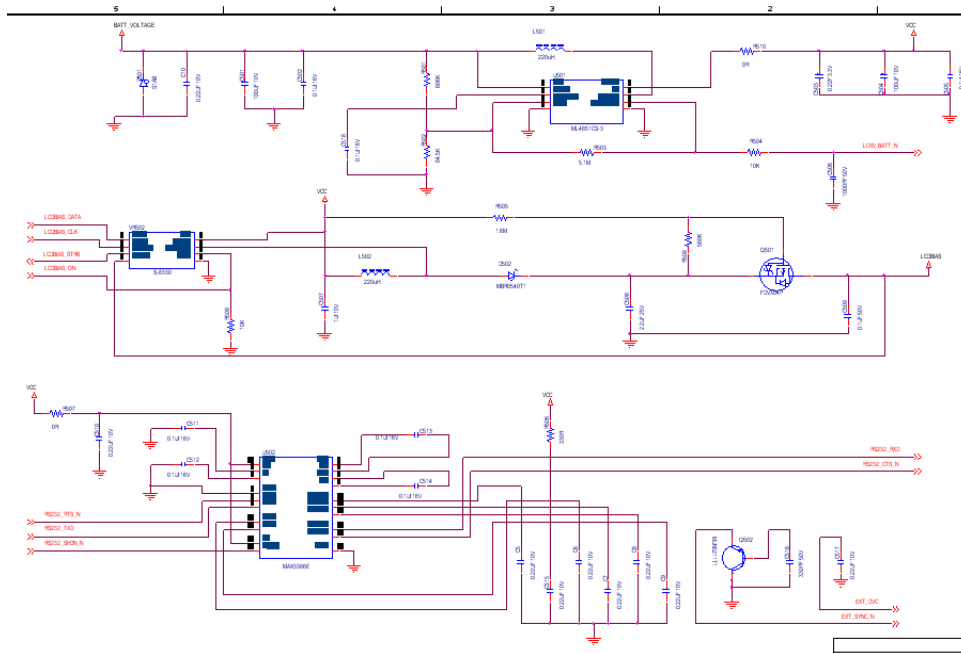
PAGINA 3



PAGINA 4



PAGINA 5



CONCLUSIONES

Durante el proceso de elaboración de esta práctica aprendí a realizar la generación de esquemáticos en la herramienta de **“Capture CSI”** y a partir de ellos crear el netlist del circuito.

La captura del esquemático así como la depuración de errores fueron relativamente sencillos, se presentaron algunos errores por el tipo de conectores que se estaban utilizando y para corregir estos errores se les cambio el tipo de conector por bidireccional, pero nos pareció mejor idea cambiar todos los conectores por off page connectors para hacer las conexiones entre páginas.

Se utilizaron varias cosas para llevar a cabo esta práctica, se hicieron análisis de las hojas de especificaciones de los componentes, y el apoyo del esquemático proporcionado por el maestro (Calvin SCH).

En la elaboración de esta práctica se detectaron varios puntos importantes (tips), y si no los tomamos en cuenta nos puede afectar en las prácticas futuras y corregirlos nos tomaría más tiempo.

CAPITULO IV. LAYOUT

OBJETIVO

Esta práctica tiene como objetivo realizar el PCB Layout tomando como referencia el archivo Calvin_SCH proporcionado por el maestro y todos los requerimientos, actividades y demás visto en clase.

El status del diseño debe de ser validado revisando símbolos, nets, shapes y DRCs.

INTRODUCCION

Un layout es un esquema de distribución de los elementos de un diseño, el cual se le presenta al cliente para vender la idea y a partir de aceptación comenzar a realizar el trabajo final en base al mismo, en otras palabras, es el boceto que se ofrece inicialmente.

METODOLOGIA

El maestro proporciono una semilla para la generación del layout, la cual ya contiene las medidas de la tarjeta y algunas configuraciones, también proporciono un placement sugerido que contenía solo algunos de los componentes más grandes, para la elaboración de esta práctica se tomaron en cuenta 3 archivos que nos ayudaron con las configuraciones necesarias y los pasos que se siguieron serán descritos más adelante:

- Primero se siguieron los pasos que vienen en el archivo Intro_to_PCB_Editor_Placement.pdf.
- Se definieron los constraints tomando en cuenta los pasos que están en el archivo Constraint_Manager_Guide.pdf.
- Se Realiza el placement tomando en cuenta el archivo Suggested_Top_Placement.pdf
- Aplicación de fan_out.
- Ruteo de las nets.
- Limpieza de DRCs.
- Se reviso el status del diseño.

PROCEDIMIENTO PARA EL DISEÑO DEL LAYOUT:

1. - Archivo Intro_to_PCB_Editor_Placement.pdf

Con la ayuda de este archivo se realizo lo siguiente:

- Se hizo la configuración de las unidades y el grid para Non-Etch y Etch
- Se hizo la configuración de “Layout Cross Section”
- Se configuro el espaciamento entre componentes
- Se realizo el QuickPlace de los componentes

Después de seguir los pasos descritos en el documento se obtuvieron los componentes del diseño colocados del lado derecho de la tarjeta., listos para ser organizados dentro de la tarjeta y continuar con el layout.

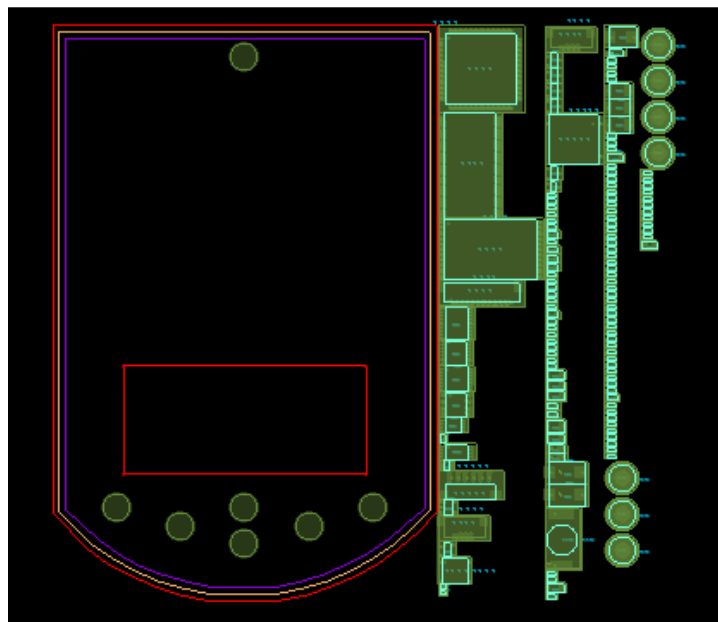


Fig.34 componentes para organizar.

2.- Archivo Constraint_Manager_Guide.pdf, para seguir los pasos definidos en este archivo se tomaron en cuenta los archivos Routing_Roules.pdf y Calvin_NetClasses_and_CSsets.pdf

Con la ayuda de estos archive se realizo lo siguiente:

- Se crearon los Csets físicos y de espaciamento.
- Se crearon Objetos, clases y Net Class-Class.
- Se asignaron Csets físicos y de espaciamento a los objetos.
- Se asignaron los constraints eléctricos.

3.- Archivo Suggested_Top_Placement.pdf

Se realizó el placement de los componentes en la tarjeta, primero se hizo el placement de los componentes grandes tomando en cuenta el placement sugerido en el archivo Suggested_Top_Placement.pdf y después se continuó con los componentes pequeños tomando en cuenta el esquemático.

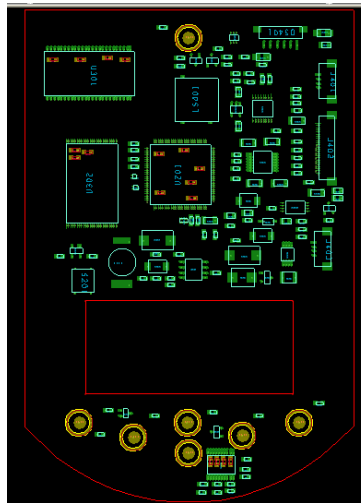


Fig.35 Placement

La mayoría de los componentes se colocaron en el layer de Top pero la posición de algunos componentes pequeños se acomodaba mejor colocándolos en el layer de Bottom.

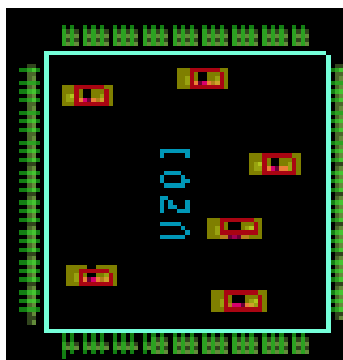


Fig.36 Componentes en Bottom.

Como se muestra en la siguiente figura(4), el componente U201 se encuentra en el layer Top y los componentes que se ven resaltados con un cuadro rojo se encuentran en el layer Bottom.

4.- Aplicación del fan_out.

Una vez que se había hecho el placement de los componentes se continuo con el fanout, para lo cual se hizo con ayuda de la herramienta, en la mayoría de los componentes la orientación se decidió tomando en cuenta hacia donde estaba dirigida su conexión pero aun así después se tuvo que hacer un ajuste manualmente.

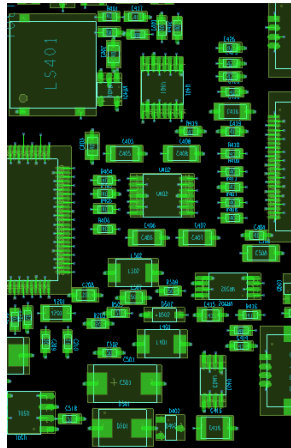


Fig.37 Fanout

5.- Ruteo de las nets.

Antes de comenzar con el ruteo se tomo en cuenta que solo se tendrían 4 layer para rutear (L3, L4, Top y Bottom), decidimos realizar el ruteo utilizando solo los layers 3 y 4 y dejar para el final los layers top y bottom, solo para casos de emergencia, es decir donde no pudiéramos pasar con los layers 3 y 4, primero se utilizaría el layer bottom y como último recurso el top.

Para comenzar con este ruteo se decidió comenzar con las señales de datos y de direcciones entre el microcontrolador y las memorias, a pesar de que se tomo la decisión de usar un layer para el ruteo horizontal y otro para el ruteo vertical, para estas señales se decidió no usar vías sino una sola conexión de un extremo al otro.

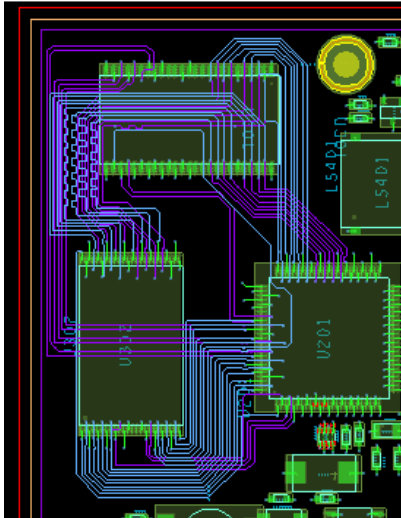


Fig.38 Ruteo de datos y direcciones

En la fig.6 se puede observar cómo se aplico un acordeón en ciertas líneas, esto debido a que se tenía que cumplir con constraints de las longitudes de línea.

Después de esto se continuó con el ruteo del par diferencial, el cristal y sus capacitores asociados hacia el microcontrolador.

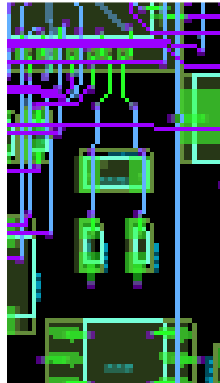


Fig.39 Ruteo del par diferencial

Se hicieron las asignaciones de los planos GND y VCC correspondientes, pero el microcontrolador necesitaba otro vcc, así que se opto por generar una isla para el mismo, dentro del plano de Vcc.

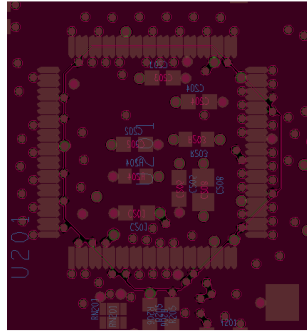


Fig.40 isla de vcc en el plano vcc

6.- Limpieza de DRCs

Durante la realización del proyecto nos encontramos con un problema, los pines 10 y 11 & 11 y 12 del microcontrolador marcaban errores de DRC (net spacing constraint L-L).

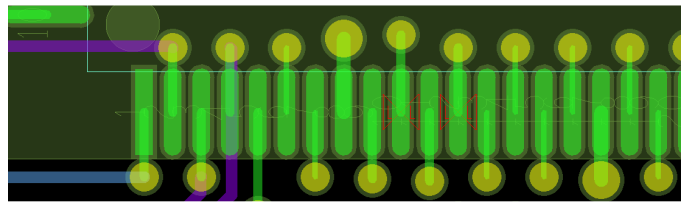


Fig.41 DRCs en pines 10y11 & 11 y 12.

Se tenía un problema entre pitch, ancho de pista y separación entre pistas, el constraint se tenía de 13.5 y el actual era de 13.18, el ancho de pista era de 6.5, $6.5 + 13.5 = 20$ y solo teníamos 19.68 de pitch, las soluciones eran hacer un poco mas angosta la pista o reducir el espacio, Como en top solo se ratearía muy poco ninguno de los casos afectaría demasiado, y si con 6.5mils de ancho de pista se logra la impedancia y con 5h se logra una buena separación dejar 4.8h no sería tan malo por una pequeña longitud.

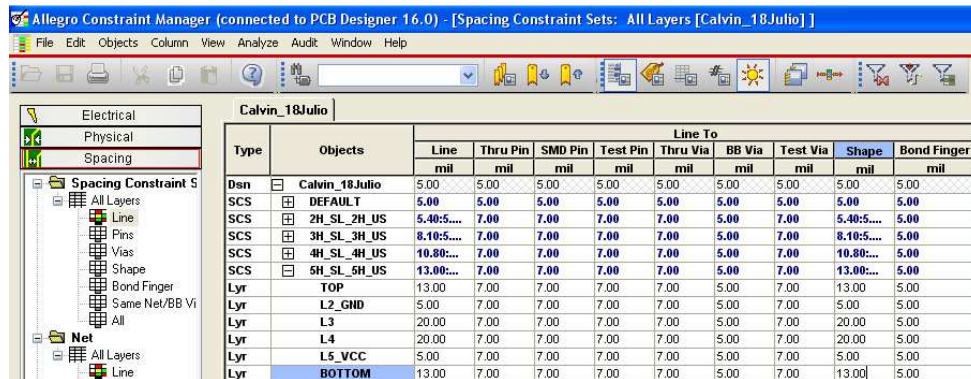


Fig.42 Cset 5h de Top

Se decidió cambiar el constraint de Top para ese Cset de 5h, se cambio de 13.5 a 13

7.- Revisión del status

Cuando se termino el ruteo del circuito se verifico que no se tuvieran nets sin rutear y que no hubiera DRCs.

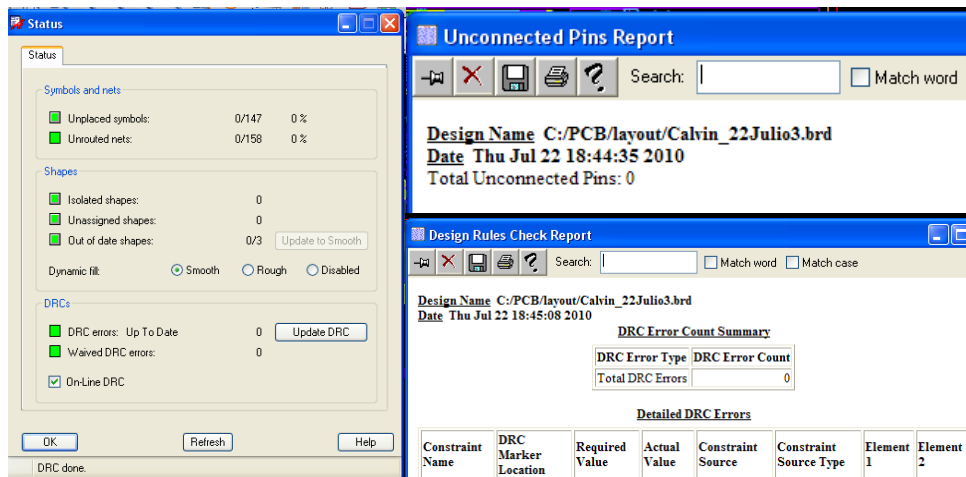


Fig.43 Revisión de status

TIPS

Se tienen algunos puntos que sería bueno tomar en cuenta durante el proceso de diseñar el layout, estos puntos podrían ahorrar tiempo en el diseño del mismo:

- Se pueden tomar solo dos layers para realizar el ruteo, se utilizo uno para el ruteo horizontal y otro para el ruteo vertical, y dejar los otros dos para emergencias como último recurso.
- Vcc y Tierra ya están conectados a un plano, pero se tiene que hacer la asignación de los mismos.
- El ruteo es más sencillo si se comienza por rutear los *nets* que cuentan con más reglas y restricciones como las señales de datos y direcciones y después realizar el ruteo de los demás componentes.

RESULTADOS

A continuación, se muestran los resultados del layout final:

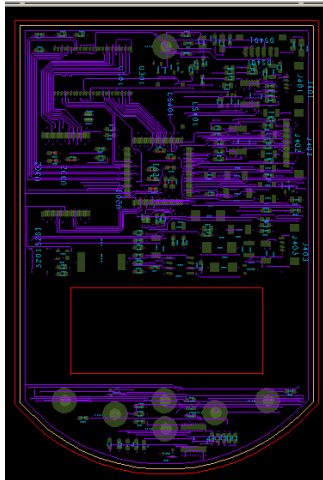


Fig.44 Layer3

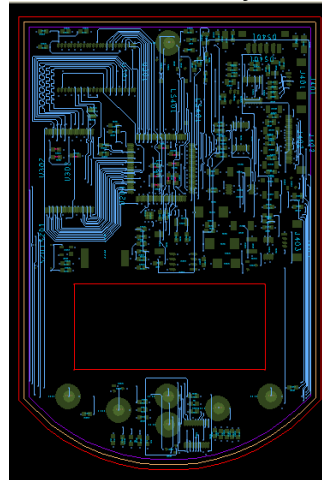


Fig.45 Layer4

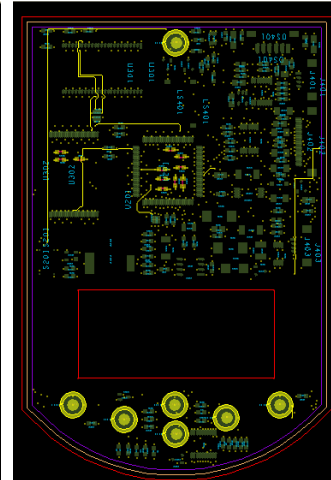


Fig.46 Layer Bottom

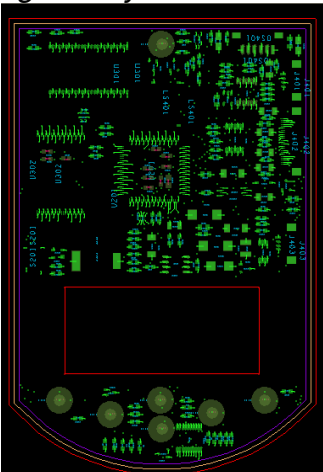


Fig.47 Layer Top

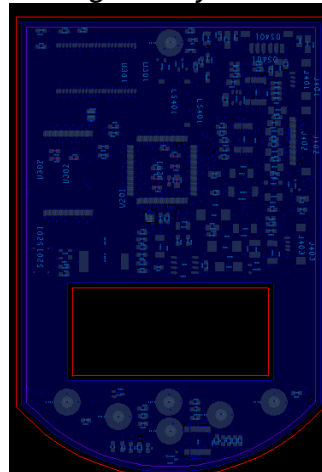


Fig.48 Plano GND

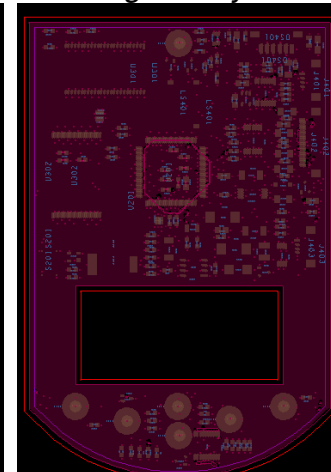


Fig.49 Plano Vcc

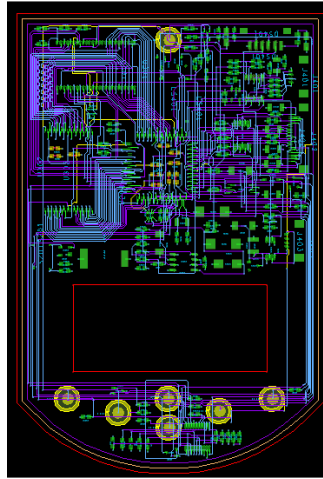


Fig.50 Layout final

CONCLUSIONES

Durante la realización de esta práctica aprendí a realizar el ruteo de nets de los componentes de una tarjeta, y que las reglas de ruteo son muy importantes así que hay que tomarlas en cuenta desde el principio.

También aprendí a agilizar el proceso declarando las reglas de ruteo en constraint manager y me di cuenta de que es muy importante determinar la posición de los componentes en la tarjeta y la manera en que se le asignarían los fanout para facilitar el ruteo.

Me di cuenta que apoyarse en el esquemático del circuito ayuda a la rápida localización de los componentes de los cuales se quiere rutear algún *net*. En general el ruteo de los *nets* de los componentes fue sencillo una vez que se había ruteado las señales de datos y direcciones.

CAPITULO V. GENERACION DE ARCHIVOS

OBJETIVO

En esta práctica se pretende llevar a cabo la realización de los archivos de fabricación de nuestro circuito usando el diseño de Layout Calvin, y revisar la correcta generación de los archivos Gerber, para esto se utilizo la herramienta de PCB editor.

METODOLOGIA

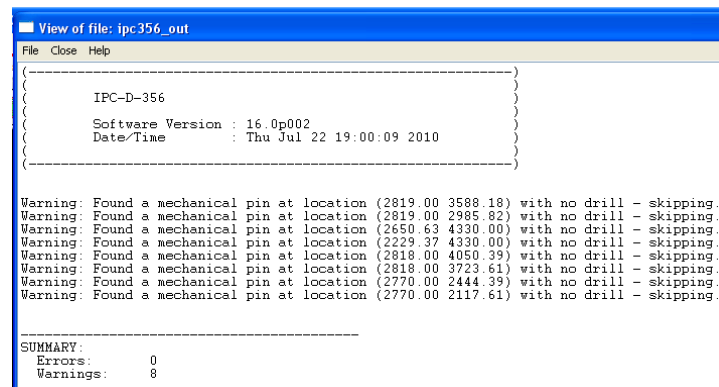
Para la elaboración de esta práctica se siguieron los siguientes pasos:

- Se generan los archivos de interconectividad de nets.
- Se generan los archivos de barrenado.
- Se genera el archivo de Gerber

PROCEDIMIENTO PARA GENERAR LOS ARCHIVOS DE FABRICACION:

1.- Generación de los archivos de interconectividad de nets.

Primero se genera el archivo de texto IPC 356, el cual se genera en File -> Export -> IPC 356, en versión se deja IPC_356^a.



```
View of file: ipc356_out
File Close Help
{
  IPC-D-356
  {
    Software Version : 16.0p002
    Date/Time       : Thu Jul 22 19:00:09 2010
  }
}

Warning: Found a mechanical pin at location (2819.00 3588.18) with no drill - skipping.
Warning: Found a mechanical pin at location (2819.00 2985.82) with no drill - skipping.
Warning: Found a mechanical pin at location (2650.63 4330.00) with no drill - skipping.
Warning: Found a mechanical pin at location (2229.37 4330.00) with no drill - skipping.
Warning: Found a mechanical pin at location (2818.00 4050.39) with no drill - skipping.
Warning: Found a mechanical pin at location (2818.00 3723.61) with no drill - skipping.
Warning: Found a mechanical pin at location (2770.00 2444.39) with no drill - skipping.
Warning: Found a mechanical pin at location (2770.00 2117.61) with no drill - skipping.

SUMMARY:
Errors:      0
Warnings:    8
```

Fig.51 Archivo de interconectividad

2.- Generación de los archivos de barrenado.

Estos archivos son una representación grafica de barrenado, Y se crean de la siguiente manera:

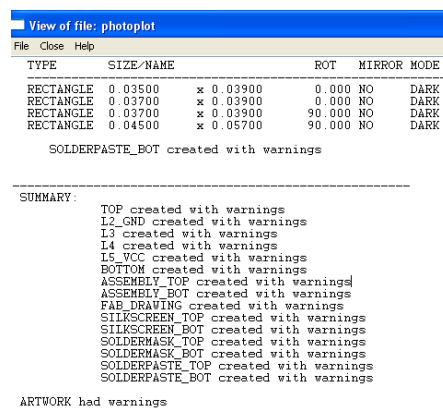
- Manufacture -> MC -> Drill customization: se sobrescriben las propiedades de las librerías(símbolo y letra)
- Manufacture -> MC -> Drill Legent: aquí se define como queremos la tabla
- Manufacture -> MC -> Parameters: se configuro a 2 enteras y 3 decimales
- Manufacture -> MC -> MC Drill: en esta ventana se tienen 2 opciones importantes, Optimize drill, que optimiza el viaje para hacer una secuencia de barrenado más directa y repeat codes, que hace el archivo más pequeño.

3.- Generación de los gerbers.

Para la creación de los gerbers se tomo en cuenta el archivo Gerber_Subclasses reference.pdf, el cual contiene una tabla en la que vienen las subclases con sus elementos para producir los gerbers.

Se modifiko el campo de Underlined line with a 6.00 para todas las subclases.

Después de generar todas las clases, se generan los gerbers seleccionando todas las clases y presionando el botón *Create Artwork*.



View of file: photoplot

TYPE	SIZE/NAME	ROT	MIRROR	MODE
RECTANGLE	0.03500 x 0.03900	0.000	NO	DARK
RECTANGLE	0.03700 x 0.03900	0.000	NO	DARK
RECTANGLE	0.03700 x 0.03900	90.000	NO	DARK
RECTANGLE	0.04500 x 0.05700	90.000	NO	DARK

SOLDERPASTE_BOT created with warnings

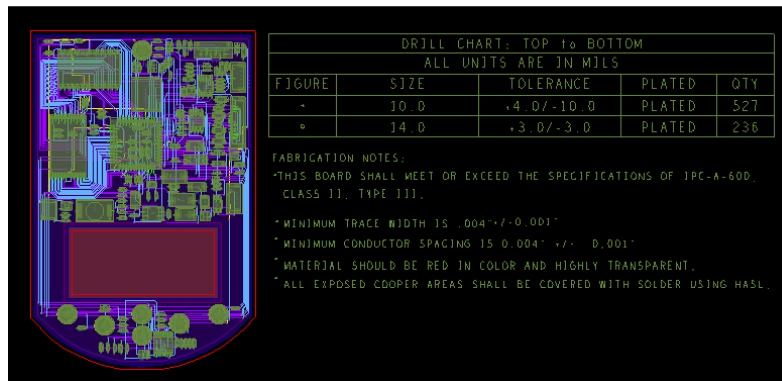
SUMMARY:

TOP created with warnings
I2_GND created with warnings
I3 created with warnings
I4 created with warnings
I5_VCC created with warnings
BOTTOM created with warnings
ASSEMBLY_TOP created with warnings
ASSEMBLY_BOT created with warnings
FAB_DRAWING created with warnings
SILKSCREEN_TOP created with warnings
SILKSCREEN_BOT created with warnings
SOLDERMASK_TOP created with warnings
SOLDERMASK_BOT created with warnings
SOLDERPASTE_TOP created with warnings
SOLDERPASTE_BOT created with warnings

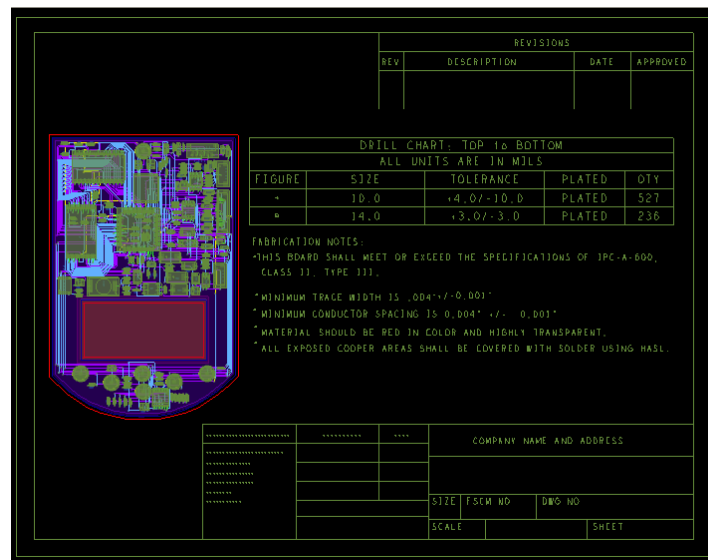
ARTWORK had warnings

Fig.52 Archivo de gerbers

Después de generar los gerbers, se agregaron 5 notas de fabricación al esquema.



Y se le dio un formato a la hoja del esquema.



TIPS

Para facilitar el proceso de la generación de subclases se tiene un método muy rápido:

- En color Dialog se apaga todo
- Se van activando las subclases que van a estar en cada grupo (datos que se tomaron de la tabla del archivo Gerber_Subclasses_reference.pdf).
- Después en file Control se selecciona la clase y se da click en Match display.

RESULTADOS

A continuación se muestran las clases, con las subclases descritas en la tabla del archivo Gerber Subclasses reference.pdf:



Fig.55 Clases y Subclases de los gerbers

CONCLUSIONES

En esta práctica fue se generaron los archivos de fabricación de nuestro circuito. Para llegar a este punto fue necesario haber concluido previamente el *Layout* del circuito. Se aprendió la manera de generar nuevos *Films* y cómo es que se pueden agregar subclases a los mismos. Se aprendió cómo generar un *Fabrication Drawing* en el cual se pueden incluir varias especificaciones del *PCB* del circuito.

En general la realización de los *gerbers* fue sencilla.

CONCLUSIONES GENERALES

Bueno, con esta práctica adquirí experiencia para identificar y llevar a cabo los principales pasos del flujo de diseño de PCBs, ya que durante este proyecto cree el layout de un sistema electrónico implementado en tecnología de tarjeta de circuito impreso, desde la creación de librerías, la captura del esquemático hasta el diseño físico del PCB mismo.

Durante la práctica de librería de símbolos aprendí a generar los símbolos de diferentes componentes, y para esto tome en cuenta varios factores como el acomodo de los pines de los componentes basándome en la hoja de especificaciones de los mismos.

Durante la práctica de librerías de empaquetado obtuve los footprints de los componentes, para esto se hizo un análisis en la hoja de especificaciones de cada componente para obtener las medidas de los pads y después de crear los pads se generaron los footprints.

Durante la práctica de la captura del esquemático aprendí a interconectar los componentes con la ayuda de wires, buses y puertos, también fue necesaria la depuración de errores y se llevo a cabo la generación del netlist del circuito.

Durante la práctica de Layout aprendí a realizar el ruteo de los nets de los componentes de un circuito en una tarjeta y a conocer reglas para aplicarlas durante el proceso de ruteo, también me di cuenta de la importancia de hacer un plan de ruteo antes de comenzar con el mismo.

Durante la práctica archivos de fabricación aprendí a generar subclases y aprendí que es importante reacomodar los identificadores de los componentes para que cuando se imprima el circuito estos datos sean legibles.

Se aplico la verificación de reglas de diseño automatizadas en tecnología de circuitos impresos, y se aplicaron reglas recomendadas de layout para manufacturabilidad, ensamble, pruebas y de diseño de layout para minimizar la interferencia electromagnética y otros efectos de alta velocidad, y por último se generaron los archivos necesarios para la fabricación del PCB.

C. PROCESAMIENTO DIGITAL DE SEÑALES

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Maestría en Diseño Electrónico

DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA



PROCESAMIENTO DIGITAL DE SENALES

Reporte de Proyecto Final

Proyecto:

Caricaturización de imágenes no sintéticas

PRESENTA:

Liliana Marcela Alvarez de la Cruz

Ana Laura Arceo Salcido

Fabio Alberto Velarde Gonzalez

Profesor: Lino Coria

Guadalajara, Jalisco. Diciembre 2008

ALFA-TOON

Cartooning non sintetic images

Alvarez De la Cruz Liliana

Iteso
Guadalajara, Mexico
skyplom@hotmail.com

Arceo Salcido Ana Laura

Iteso
Guadalajara, Mexico
silvercrowley@hotmail.com

Velarde Gonzalez Fabio Alberto

Iteso
Guadalajara, Mexico
fabio.velarde@hotmail.com

Abstract— In this paper we present an algorithm for non-photorealistic rendering. We introduce an algorithm that produces an image that gives the impression that is a hand-made cartoon version of the original image. We present a method that divides the problem into three categories (color quantization, edge detection and edge distortion) in order to achieve our goal.

1. Introduction

Entertainment is one of the most common uses for a personal computer these days and has pushed the growth of the computer industry. We can think in a long list of fun things that we can do with a computer, we can listen to music, watch a movie or play a videogame, but there is another kind of entertainment, one that is not only funny but it can help people get more involved with a computer.

With digital imaging it is very easy to take a picture and uploaded it to a computer, and once the picture is in the computer, you can have all the fun you want with it. In this matter, one form of entertainment that can achieve the two objectives mentioned earlier is the idea of being able to turn a personal photograph into a cartoon.

Non-photorealistic rendering provides a way of entertainment and has been developing over the last years, and it is very easy to find software that provides this kind of image processing. in this paper we introduces an algorithm that provides a new approach to this kind of image processing, where we focus on give a "hand made" effect to the image we produce

2. Algorithms

2.1. COLOR REDUCCION

To reduce the colors in an image is used a process called quantization, this algorithm reduces the colors of

an image, and this process usually consists of three steps [1]:

- 1).-it makes groups with the representative colors
- 2).-It is a palette with the average values of each group
- 3).-finally replaced the values of colors of a palette to the original image.

The algorithm that was implemented is the show it below and consists of three steps:

To reduce the colors of the original image (Figure 1), first, it makes a reduction in cromancy (Figure 3a).



Figure 1. Original image (photo)

The second step was made with the palette of the figure 2. Each color in the palette has a value in the three components R, G and B., this process reduces the first quantized image to every possible combination with the values 51, 102, 153, 204 y 255 (Figure 3b).

The third step was, it defines 22 colors that will use to the cartoon, and grouping all the resultant colors in the step 2, within them. It was replaced the 22 colors in the resulting image of the step 2(Figure 3c).

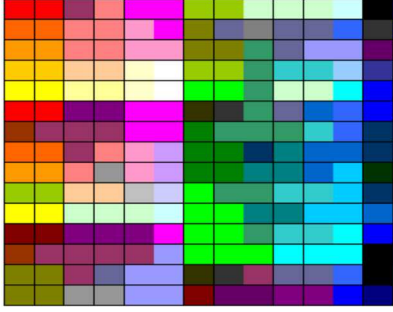


Figure 2. Color palette

The results of quantization are showing at the figure 3. a) First step b) second step and c) third step.

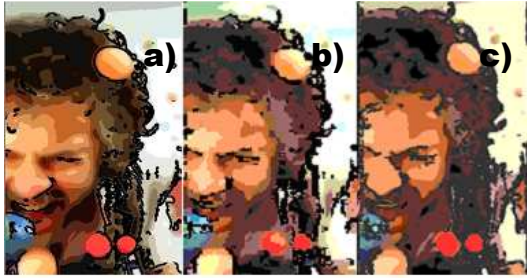


Figure 3. Quantization

2.2. EDGES DETECTION

In a cartoon it is very important to be able to distinguish between different contour lines. A contour that have more importance than other needs to be drawn different, with a thicker line. To do so, we did and wavelet analysis. Using the `dwt2` function of *matlab* we obtain four different matrixes (figure 4)

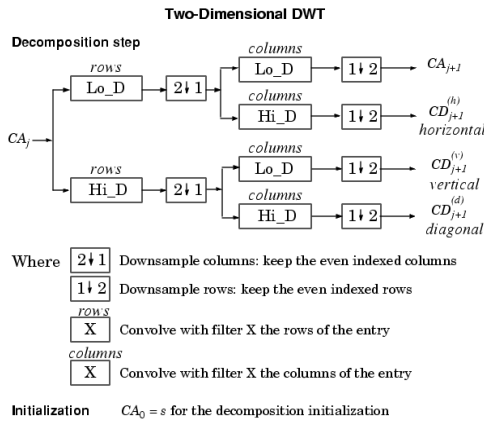


Figure 4. Two dimensional dwt

The $CD^{(d)}$ matrix provides all information for that is relevant and important for both the rows and the columns in the images, $CD^{(h)}$ provides this information that is only relevant in the rows and $CD^{(v)}$ for the columns.

We detected the edges of all these matrixes, and we re-draw them with different level of thinness for each one of the matrixes. With this, edges detected in both rows and columns will be thicker than those that were detected in only one of these different scopes.

2.3. EDGES DISTORTION

When an image is caricatured it is necessary to give an effect to the edges in order that they seem to be handmade for that we used a filter called motion.

The Motion Blur filter creates a movement blur. The filter is capable of Linear, Radial, and Zoom movements. Each of these movements can be further adjusted, with Length, or Angle settings available [3].

The filter used cause a linear motion effect.

Linear Motion Filter

Is a blur that travels in a single direction, horizontally, for example. In this case, Length means as Radius in other filter, it represents the blur intensity. More Length will result in more blurring. Angle describes the actual angle of the movement. Thus, a setting of 90 will produce a vertical blur, and a setting of 0 will produce a horizontal blur.

The following describes the function used in *matlab* to give the effect of handmade in the image:

`h = fspecial('motion', len, theta)` returns a filter to approximate, once convolved with an image, the linear motion of a camera by `len` pixels, with an angle of `theta` degrees in a counterclockwise direction. The filter becomes a vector for horizontal and vertical motions. The default `len` is 9 and the default `theta` is 0, which corresponds to a horizontal motion of nine pixels.

Motion filter Algorithm[4]:

To compute the filter coefficients, `h`, for 'motion':

1) Construct an ideal line segment with the desired length and angle, centered at the center coefficient of `h`.

2) For each coefficient location (i, j), compute the nearest distance between that location and the ideal line segment.

3) $h = \max(1 - \text{nearest_distance}, 0)$;

4) Normalize h: $h = h / (\sum(h(:)))$

Are observed the edges detected of the real image (figure 4b)



Figure 5. Original image

After filtering the image, we observe the effect that the filter cause to the edges (figure 5b)



Figure 6. Distortion

3. RESULTS

The following figures show the results achieved with ALFA-TOON.



Figure 7. Test image 1



Figure 8. Test image 2



Figure 9. Test image 3

The images below show a comparison between a public program (Figure 9b) in the web [5] and ALFA-TOON (Figure 9a)



Figure 10. Comparison

The results of ALFA-TOON can valuate only subjectively, then to evaluate ALFA-TOON was carried out a survey in which we show them a picture caricatured and they assigned us a qualification between 1 and 10 and we obtained an average of 8.

References

- [1] http://studies.ac.upc.edu/EPSC/TCP/documentos/Enunciado_Cuantizador_0506Q1.doc
- [2] <http://platea.pntic.mec.es/~abercian/guiahtml/colores.html>
- [3] <http://docts.gimp.org/en/plugin-in-mblur.html>
- [4] <http://mathworks.com/access/helpdesk/help/toolbox/images/index.html?access/helpdesk/help/toolbox/images/fspecial.html>
- [5] <http://www.befunky.com>

